

UNIVERSIDAD CARLOS III DE MADRID

TRABAJO FIN DE GRADO



***SIMULACIÓN DE UN BRAZO ROBÓTICO EN
SIMMECHANICS***

GRADO EN INGENIERÍA ELECTRÓNICA Y AUTOMÁTICA
INDUSTRIAL

Autor: D. Antonio Rodríguez Ramos

Tutor: D. Dorin Sabin Copaci

Leganés, Octubre de 2015





AGRADECIMIENTOS

A mis padres, Miguel y Antonia, que me lo han dado todo.

A ti, abuela, por toda tu fuerza.

A mis hermanos, fuente de apoyo constante.

A mi familia y amigos, por sus ánimos a lo largo de todos estos años.

Y a mis tutores, David y Dorin, gracias por vuestra ayuda y paciencia.

Por ti, por ellos, por mí. GRACIAS.



RESUMEN

El MANFRED es uno de los proyectos desarrollados por el departamento de Ingeniería de Sistemas y Automática de la Universidad Carlos III de Madrid y está focalizado en el objetivo de desarrollar manipuladores móviles avanzados capaces de operar en entornos humanos. Actualmente este estudio se encuentra en la tercera generación de su desarrollo, conocida como MANFRED 3.

Dentro de este proyecto se enmarca el presente Trabajo Fin de Grado. El objetivo de este estudio es la creación de un entorno de simulación fiable para el manipulador MANFRED 3, con el fin de realizar pruebas de trayectorias y analizar los pares asociados a los motores que componen el brazo articulado del manipulador, obteniendo de este modo una mejor comprensión de su comportamiento.

Con el fin de crear el entorno virtual de simulación se han realizado las siguientes tareas:

- Diseño e implementación del modelo dinámico inverso del manipulador, a través de SimMechanics.
- Elaboración del modelo cinemático del brazo robótico haciendo uso de los parámetros de Denavit-Hartenberg que definen al manipulador. De este modelo se obtienen el conjunto de valores de posición, velocidad y aceleración asociados a cada articulación del robot, los cuales servirán como entrada para el modelo generado en SimMechanics.
- Simulación del MANFRED 3, identificando las posibles desviaciones o errores entre la generación de ambos subsistemas e intentando corregirlos, así como obteniendo simulaciones gráficas de los mismos.

Finalmente, en el apartado de conclusiones se describen los puntos de interés encontrados a lo largo del desarrollo del estudio, además de ofrecer futuras acciones de mejora o de continuación del proyecto.

ABSTRACT

MANFRED is one of the different projects where the department of Automatics and Systems Engineering of University Carlos III of Madrid is involved, and it is focused on the development of mobile manipulators that are able to operate in human environments. This study is currently in the third generation of development, known as MANFRED 3.

The objective of the present study is to create a reliable simulation environment for the manipulator MANFRED 3 in order to perform trajectories tests and analyze associated pairs engines that comprises the articulated manipulator arm, thus obtaining a better understanding their behavior.

Or order to create the virtual environment simulation the following tasks have been performed:

- Design and implementation of the inverse dynamic model of the manipulator, through SimMechanics.
- Preparation of the kinematic model of the robotic arm using Denavit-Hartenberg parameters that define the manipulator. The values of position, velocities and acceleration related to the model are obtained for each joint of the robot in order to be used as the inputs for the SimMechanics model.
- Simulation of MANFRED 3 identifying any deviations or errors between the generation of both subsystems and trying to fix them. Besides graphical simulations will be obtained.

Finally, the conclusions of the study are detailed in chapter 5, collecting the remarkable points founded during the simulation and where new improvements are suggested.



ÍNDICE

RESUMEN.....	IV
ABSTRACT	V
ÍNDICE.....	VI
ÍNDICE DE FIGURAS.....	VIII
ÍNDICE DE TABLAS	X
LISTADO DE ACRÓNIMOS	XI
1 INTRODUCCIÓN	2
1.1 MANFRED 3ª GENERACIÓN	5
1.2 OBJETIVO.....	6
1.3 ESTRUCTURA DEL PROYECTO	7
1.4 INTERÉS DEL AUTOR	7
2 HERRAMIENTAS SW	8
2.1 SOLIDWORKS.....	8
2.2 MATLAB.....	8
2.2.1 SIMMECHANICS.....	9
2.2.2 SIMMECHANICS LINK.....	9
2.2.3 MATLAB-PETER CORKE TOOLBOX	9
3 DESARROLLO DEL ESTUDIO.....	10
3.1 DESCRIPCIÓN GENERAL	10
3.2 BLOQUE CINEMÁTICO INVERSO.....	11
3.2.1 HERRAMIENTA PETER CORKE	12
3.3 BLOQUE DINÁMICO INVERSO	23
3.3.1 SOLIDWORKS.....	23
3.3.2 SIMMECHANICS.....	26
3.3.3 CONEXIONADO DE BLOQUES.....	29
3.4 AJUSTES DEL ENTORNO DE SIMULACIÓN.....	30
4 PRUEBAS DE SIMULACIÓN	33
4.1 PRUEBAS I: TRAYECTORIAS SIMPLES SOBRE CADA EJE	34
4.2 PRUEBAS II: TRAYECTORIAS COMBINADAS	38
4.3 PRUEBAS III: TRAYECTORIA CIRCULAR.....	45
5 CONCLUSIONES	50
BIBLIOGRAFÍA.....	51
ANEXOS	52
ANEXO I: PRESUPUESTO	53
DESGLOSE DE HORAS.....	53
COSTES DE MATERIAL.....	54
COSTES DE PERSONAL	55
COSTE TOTAL.....	56
ANEXO II: PROGRAMACIÓN DEL BRAZO ANTROPOMÓRFICO	57
CÓDIGO DE DEFINICIÓN DEL BRAZO	57
OTROS COMANDOS ÚTILES.....	57



ANEXO III: PIEZAS SOLIDWORKS	58
ANEXO IV: ESQUEMAS DE ENSAMBLAJES	611

ÍNDICE DE FIGURAS

FIGURA 1 ESQUEMA GENERAL BRAZO HUMANO-ROBOT [11]	2
FIGURA 2 LWR III, DLR	3
FIGURA 3 IIWA, KUKA	3
FIGURA 4 JACO ² , KINOVA	4
FIGURA 5 LWA 4D, ROBOTNIK	4
FIGURA 6 DIMENSIONES FINALES DEL MANFRED	5
FIGURA 7 ENSAMBLAJE FINAL	6
FIGURA 8 ESQUEMA DESCRIPTIVO DEL ENTORNO DE SIMULACIÓN	10
FIGURA 9 DEFINICIÓN ROBOT 2-LINKS CON EL MANIPULADOR, [7]	12
FIGURA 10 REPRESENTACIÓN GRÁFICA DEL MANIPULADOR	13
FIGURA 11 MODIFICACIÓN POSICIÓN DEL MANIPULADOR.....	13
FIGURA 12 BLOQUES ASOCIADOS AL BRAZO	14
FIGURA 13 ESQUEMA CÁLCULO MOVIMIENTO-JACOBIANA EXPLICATIVO, [7]	16
FIGURA 14 ESQUEMA CÁLCULO VELOCIDAD ARTICULAR	17
FIGURA 15 DISCRETE-INTEGRATOR: FIJADO DE POSICIÓN INICIAL	18
FIGURA 16 BLOQUE JTRAJ	19
FIGURA 17 ESQUEMA CINEMÁTICO INVERSO MEDIANTE JTRAJ	20
FIGURA 18 MODELO CINEMÁTICO INVERSO FINAL	20
FIGURA 19 BLOQUE CINEMÁTICA DIRECTA DEL BRAZO	21
FIGURA 20 BLOQUES LIBRERÍA <i>TOOLBOX</i> PARA TRANSFORMADA HOMOGÉNEA.....	21
FIGURA 21 CINEMÁTICA DIRECTA	22
FIGURA 22 SITUACIÓN INICIAL MANIPULADOR MANFRED 3.....	23
FIGURA 23 EVOLUCIÓN ESTRUCTURA. ESTADO INICIAL, SIMETRÍA Y ESTADO FINAL	24
FIGURA 24 MANFRED 3, SITUACIÓN FINAL CON TAPAS	25
FIGURA 25 <i>ROOTGROUND</i>	26
FIGURA 26 <i>WELD</i>	26
FIGURA 27 <i>BODY</i>	27
FIGURA 28 <i>REVOLUTE</i>	27
FIGURA 29 ESQUEMA SIMMECHANICS TRAS MODIFICACIÓN INICIAL	28
FIGURA 30 BLOQUE <i>DEMUX</i>	29
FIGURA 31 CONEXIÓN PARÁMETROS A SIMMECHANICS	29
FIGURA 32 ENTORNOS GRÁFICOS PETER CORKE -SIMMECHANICS.....	30
ILUSTRACIÓN 33 AJUSTE SALIDA BLOQUE DINÁMICO INVERSO (POSICIÓN CARTESIANA EXTREMO ROBOT)	31
ILUSTRACIÓN 34 ERROR ENCONTRADO DURANTE AJUSTES DE SIMULACIÓN.....	32
FIGURA 35 PI-A-SALIDA IK (m/s)	34
FIGURA 36 PI-A-SALIDA ID (m/s)	34
FIGURA 37 PI-A-DIFERENCIA ENTRE SALIDAS	35
FIGURA 38 PI-A-TRAYECTORIA DIBUJADA IK vs.ID (m)	35
FIGURA 39 PI-B-SALIDA BLOQUE IK (m/s)	36
FIGURA 40 PI-B-SALIDA BLOQUE ID (m/s)	36
FIGURA 41 PI-B-DIFERENCIA SALIDAS (m/s)	37
FIGURA 42 PI-B-TRAYECTORIA DIBUJADA IK vs ID(m)	37
FIGURA 43 PII-A-SALIDA BLOQUE IK (m/s).....	38
FIGURA 44 PII-A-SALIDA BLOQUE ID (m/s)	38
FIGURA 45 PII-A-DIFERENCIA DE SALIDAS (m/s)	39
FIGURA 46 PII-A-TRAYECTORIA DIBUJADA IK vs. ID (m).....	39
FIGURA 47 PII-B-SALIDA BLOQUE IK (m/s).....	40
FIGURA 48 PII-B-SALIDA BLOQUE ID (m/s)	40
FIGURA 49 PII-B-DIFERENCIA SALIDAS (m/s)	41
FIGURA 50 PII-B-TRAYECTORIA DIBUJADA HASTA SEGUNDO 24 IK vs. ID (m).....	41
FIGURA 51 PII-B-TRAYECTORIA DIBUJADA TRAS SEGUNDO 24 IK vs ID(m).....	42



FIGURA 52 PII-C-SALIDA BLOQUE IK (m/s)	43
FIGURA 53 PII-C-SALIDA BLOQUE ID (m/s)	43
FIGURA 54 PII-C-DIFERENCIA ENTRE SALIDAS (m/s)	44
FIGURA 55 PII-C-TRAYECTORIA DIBUJADA IK VS.ID (m)	44
FIGURA 56 POSICIÓN INICIAL DEL ROBOT PREVIO DIBUJO DE CÍRCULO.....	45
FIGURA 57 PIII- BLOQUE IK (m/s)	46
FIGURA 58 PIII-BLOQUE ID (m/s).....	46
FIGURA 59 DIFERENCIA SALIDAS (m/s).....	47
FIGURA 60 TRAYECTORIAS DESCRITAS POR EL BRAZO IK VS.ID	47
FIGURA 60 PAR MOTOR OBTENIDO	48
FIGURA 62 12012_ARQ_M3_A003_P_19_00	59
FIGURA 63 DUMMY_J1F_12.....	59
FIGURA 64 DUMMY_J2F_12.....	59
FIGURA 65 DUMMY_J3F_12.....	59
FIGURA 66 12102_ARQ_M3_A003_P_00J1	59
FIGURA 67 12102_ARQ_M3_A003_P_00J2	59
FIGURA 68 12102_ARQ_M3_A003_P_00D2	59
FIGURA 69 12102_ARQ_M3_A003_P_00J3	60
FIGURA 70 12102_ARQ_M3_A003_P_00D1.....	60
FIGURA 71 12102_ARQ_M3_A003_P_00D3.....	60
FIGURA 72 CÍRCULO.....	62
FIGURA 73 ESQUEMA FINAL DEL ENTORNO DE SIMULACIÓN.....	63



ÍNDICE DE TABLAS

TABLA 1 PRINCIPALES CARACTERÍSTICAS DEL MANFRED 3,[1]	5
TABLA 2 PARÁMETROS DH.....	11
TABLA 3 COSTES DE MATERIAL	54
TABLA 4 COSTES DE PERSONAL	55
TABLA 5 COSTE TOTAL	56

LISTADO DE ACRÓNIMOS

CAD	Computer Aided Design (Diseño asistido por computador)
DH	Denavit Hartenberg
FK	Forward Kinematics (Cinemática Directa)
GDL	Grados de Libertad
ID	Inverse Dynamics
IK	Inverse Kinematics (Cinemática Inversa)
LWR	Light Weight Robot
MANFRED	MAN FRiEnDly mobile manipulator
SW	Software
TFG	Trabajo Fin de Grado
UC3M	Universidad Carlos III de Madrid



1 INTRODUCCIÓN

Un robot antropomórfico es un sistema cuya finalidad es tener un comportamiento similar al elemento humano que se busca imitar.

Si existe un elemento del cuerpo humano cuyo equivalente robótico se lleva desarrollando desde hace años ese es el brazo. Un esquema general de esta equivalencia puede encontrarse en la siguiente figura.

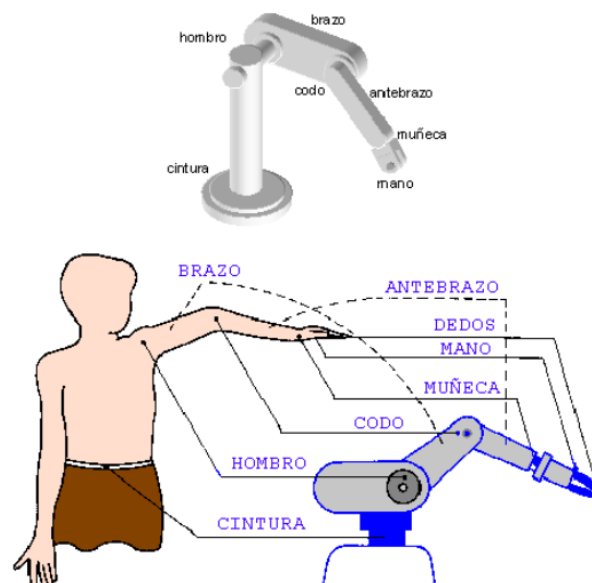


Figura 1 Esquema general brazo humano-robot [11]

Generalmente, un robot antropomórfico, también conocido como robot de codo o robot angular, suele presentar una configuración basada en 3 articulaciones de posicionamiento y 3 de orientación, aunque

no es una configuración fija. Gracias a ello, estos sistemas ofrecen una gran maniobrabilidad y accesibilidad, además de ser muy rápidos y capaces de realizar trayectorias complejas.

Estas características han hecho que los brazos robóticos antropomórficos lleven décadas implantados en la industria con el objetivo de mejorar la calidad y la producción o bien con el fin de trabajar en ambientes peligrosos y nocivos para el hombre.

Sin embargo, en la actualidad, el reto de estos sistemas es el de incorporarse a la vida cotidiana fuera de un entorno industrial. Para ello se necesitan robots que sean capaces de realizar tareas de manera autónoma, moverse en entornos humanos, interactuar con personas, manipular objetos y todo ello cumpliendo con elevados criterios de robustez y seguridad física.

Algunos de los manipuladores más avanzados que existen en la actualidad se muestran a continuación:

- LWR III de DLR:

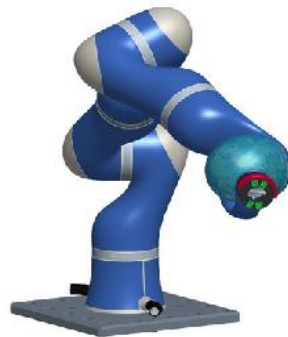


Figura 2 LWR III, DLR

La tercera generación de este robot, con un peso del brazo de 13,5 Kg y capaz de manejar cargas de hasta 15 kg. Con este brazo DLR ha conseguido crear un robot de 7 GDL con una carga de relación en peso mayor de 1:1.

- LBR IIWA de KUKA ROBOTICS:



Figura 3 IIWA, KUKA

KUKA Roboter GmbH es uno de los fabricantes principales de robots industriales. Es un brazo de 7 DOF y con el que según a la descripción del fabricante se ha buscado abrir nuevas zonas de aplicación en el entorno humano que estaban cerradas a los robots.

- KINOVA JACO²:



Figura 4 JACO², KINOVA

El brazo Jaco2 de Kinova es un robot de 6 GDL. De acuerdo con los datos del fabricante con este brazo se consigue que los usuarios interactúen con su entorno con total eficacia y libertad.

- LWA 4D de ROBOTNIK:



Figura 5 LWA 4D, ROBOTNIK

Brazo de 7 GDL con un peso de 18 kg y una capacidad de carga de 10kg creado por la empresa ROBOTNIK.

Además de este conjunto de manipuladores creados por distintos fabricantes industriales, numerosas Universidades y equipos de investigación están también inmersos en el estudio y desarrollo de estos manipuladores orientados a realizar su actividad en entornos cotidianos para el hombre fuera del ámbito industrial.

Dentro de este marco se encuentra el proyecto MANFRED, manipulador móvil desarrollado por el Grupo de Manipuladores Móviles de RoboticsLab, del Departamento de Ingeniería de Sistemas y Automática de la Universidad Carlos III de Madrid,[10]

1.1 MANFRED 3ª GENERACIÓN

El proyecto MANFRED, [2] [10] está enfocado a prestar ayuda a las personas con sus tareas diarias, siempre dentro de un entorno viable y seguro donde se requieran funciones similares a las que son habitualmente realizadas por un brazo humano (por ejemplo: casa, hospital, oficinas...).

Basado en el diseño de las generaciones anteriores del manipulador y en el *know-how* adquirido en proyectos similares, el equipo de Robotics Lab, [9] ha creado la tercera generación del MANFRED. Entre otras, el nuevo brazo tiene las siguientes particularidades:

- Brazo con 7 GDL.
- Dimensiones: 998.5 milímetros.
- Capacidad de carga: 7 kilogramos .

Características que son similares a las de un brazo humano y con solo un peso del brazo antropomórfico de 14 Kilogramos.

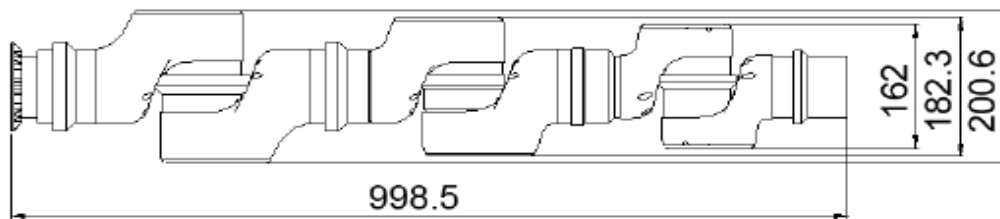


Figura 6 Dimensiones finales del MANFRED

Tabla 1 Principales características del MANFRED 3,[1]

PARAMETROS	DESCRIPCIÓN
Dimensiones generales	900-1000 mm de longitud del hombro a la muñeca 200 mm de máximo diámetro externo
Grados de Libertad (GDL)	7 GDL (en total) R-P-R-P-R-P-R (donde R es roll y P es Pitch)
Masa	≤ 14 kg
Carga útil	≥ 50 N
Velocidad de rotación	$\leq 180^\circ/s$ para cada articulación
Precisión	1 mm
Repetibilidad	± 0.05 mm
Sensores	Posición ,Torque ,Frecuencia,Consumo
Voltaje de alimentación	48 VDC
Cableado	El Sistema de cableado debe ir dentro del brazo robótico

Con este nuevo brazo se busca prestar atención a los siguientes aspectos de acuerdo con la descripción que se hace en el documento del diseño[1]:

- Reducir su peso (de 18 kg en la generación anterior a 14 kg).
- Añadir un GDL extra (de 6 a 7, con el fin de resolver los problemas asociados con la no-redundancia: singularidades y espacios de trabajo obstruidos).
- Distribuir independientemente una fuerza y control de posición en cada articulación (para evaluar de manera apropiada el desempeño del brazo).
- Incluir todo el sistema de cableado dentro del brazo (junto con su electrónica), buscando una mejora en la seguridad de usuarios potenciales y reducir el riesgo asociado al posible estrangulamiento de los elementos.
- Instalar el brazo en la plataforma móvil del MANFRED con la finalidad de completar el diseño antropomórfico del manipulador con dos brazos.



Figura 7 Ensamblaje final

1.2 OBJETIVO

El principal objetivo del presente proyecto es la generación de una simulación del manipulador robótico MANFRED 3ª generación.

Mediante esta simulación se pretende obtener un entorno fiable y preciso con la finalidad de que sirva de base a partir de la cual se puedan realizar, con posterioridad, distintos estudios del comportamiento del brazo robótico.

El conjunto de tareas a realizar para la consecución de este objetivo serán las siguientes:

- Reconstrucción del brazo robótico mediante la herramienta CAD SolidWorks 2014.
- Obtención del diagrama de bloques en SimMechanics asociado al diseño del manipulador.
- Generación del modelo cinemático del brazo antropomórfico mediante el uso de la *toolbox* de Peter Corke,[6]. De este modelo cinemático se obtendrán los valores de posición, velocidad y aceleración de los motores del robot para una trayectoria dada.

- Unificación del modelo cinemático generado con la *toolbox* y del modelo dinámico generado en SimMechanics.
- Corrección del modelo unificado mediante la agregación de sensores de posición y ajuste de offset.
- Ajustado el entorno de simulación se determinarán distintos movimientos a realizar por el robot con los que comprobar la fiabilidad y utilidad del modelo generado.

1.3 ESTRUCTURA DEL PROYECTO

Este proyecto consta de un total de cinco capítulos, siendo el primero de ellos la presente introducción donde quedan acotados los objetivos generales de este estudio.

A continuación se procede a describir la actividad a desarrollar en cada uno de ellos:

- Capítulo 1, Introducción: en este capítulo se realiza una pequeña introducción para situar al lector en el entorno en el que se desarrolla este proyecto.
- Capítulo 2, Herramientas SW: en este capítulo se expone una breve explicación de las herramientas utilizadas a lo largo de la simulación.
- Capítulo 3, Desarrollo del estudio: este es el capítulo central del proyecto, donde se exponen el conjunto de trabajos realizados en cada uno de los bloques de modelado que conforman simulación. Además, resulta de especial importancia el apartado dedicado a los ajustes de errores.
- Capítulo 4, Desarrollo de la simulación: pruebas de simulación para comprobar la efectividad del entorno generado.
- Capítulo 5, Conclusiones y posibles trabajos a realizar en el futuro: en este apartado se presentan las conclusiones principales obtenidas de la realización del proyecto, así como los posibles trabajos que puedan suponer una evolución de lo desarrollado en el mismo.
- Anexos: en este apartado se hará un desglose de los anexos útiles para el desarrollo del estudio, donde se incluye información presupuestaria del proyecto, esquemas de piezas y diagramas.

1.4 INTERÉS DEL AUTOR

La motivación nace de la posibilidad de aplicar parte de los conocimientos adquiridos a lo largo del desarrollo de los estudios de la especialidad en el Grado en Ingeniería Electrónica Industrial y Automática, además del interés por conocer y profundizar en los conceptos relacionados con la robótica, teniendo la oportunidad de ser una pequeña parte en el desarrollo de una aplicación real.

2 HERRAMIENTAS SW

Para la realización de este proyecto es necesario el uso de un conjunto de herramientas software con las que realizar la simulación objeto de este estudio. En este capítulo se encuentra una breve descripción de cada una de las herramientas utilizadas.

2.1 SOLIDWORKS

SolidWorks es un software de diseño asistido por computadora que se usa para el modelado mecánico en 3D.

Con esta herramienta se modelan las piezas que constituyen el manipulador. Además de poder crear las piezas virtualmente, este software también permite caracterizarlas o lo que es lo mismo, darle las propiedades que tendría esa pieza en la realidad, como son su masa, centro de masa o momento de inercia.

Del mismo modo, una vez generadas todas las piezas con esta herramienta también se crea el ensamblaje de éstas, relacionando una a una todas las piezas hasta tener el manipulador.

La última opción a aprovechar de la herramienta es la de instalarle el complemento de SimMechanics. Mediante esta opción una vez modelado el ensamblaje del manipulador podremos exportarlo a un archivo con formato de lectura de Matlab-SimuLink (apartado siguiente). Desde modo se podrá obtener directamente el modelo dinámico inverso del manipulador.

2.2 MATLAB

Matlab, abreviatura de MATrix Laboratory (en español “laboratorio de matrices”) es una herramienta de software matemático, que ofrece un entorno de desarrollo integrado y con un lenguaje propio de programación.

Para el desarrollo del presente proyecto se utilizará, sobre todo, uno de los productos que está integrado en la herramienta Matlab. De acuerdo con descripción,[4] , Simulink es un entorno de diagramas de bloque para la simulación multidominio y el diseño basado en modelos. Admite el diseño y simulación a nivel de sistema, la generación automática de código y la prueba y verificación continua de los sistemas embebidos.

Este software será muy importante para el desarrollo del proyecto, pues será la base que sustentará el modelado y la simulación del sistema diseñado.

Dentro del entorno Matlab-Simulink se puede hacer una subdivisión acorde a las *toolbox* o entornos que se han agregado.

2.2.1 SIMMECHANICS

SimMechanics, [5] proporciona un entorno de simulación multicuerpo para sistemas 3D mecánicos, tales como robots, suspensiones de vehículos, equipos de construcción, y el tren de aterrizaje del avión. Permite modelar el sistema usando bloques que representan los órganos, articulaciones, restricciones y elementos de fuerza y, tras ello, SimMechanics formula y resuelve las ecuaciones de movimiento para el sistema mecánico completo. Modelos de sistemas CAD, incluyendo la masa, la inercia, las articulaciones, limitación, y la geometría 3D, se pueden importar en SimMechanics. Una animación 3D generada automáticamente le permite visualizar la dinámica del sistema.

En el ámbito de este proyecto, se utilizará SimMechanics para importar el sistema generado con SolidWorks creando el conjunto de bloques que representen el modelo dinámico inverso del manipulador. Posteriormente se generará la animación 3D para visualizar la simulación dinámica objeto de este estudio.

2.2.2 SIMMECHANICS LINK

SimMechanics Link se utiliza para instalar en SolidWorks la herramienta complemento que permite a éste exportar modelados CAD interpretables para el entorno de SimMechanics, generando así el diagrama de bloques en SimMechanics al modelo generado con la herramienta de modelado mecánico.

2.2.3 MATLAB-PETER CORKE TOOLBOX

La *toolbox* de Peter Corke (cuyo nombre se debe a su creador) al igual que toda la información relativa a esta herramienta SW puede encontrarse en la web,[6]. La función de esta *toolbox* es realizar simulaciones de brazos robóticos, útil por tanto para el desarrollo del presente estudio, ya que permite generar mediante métodos generales las trayectorias, la cinemática o la dinámica de los brazos.

Su utilización en este estudio estará enfocada a generar el modelo cinemático del brazo, a partir del que se obtendrán los valores articulares de posición, velocidad y aceleración que se tomarán como entrada para el diagrama de bloques de SimMechanics. De este modo, la simulación en SimMechanics ofrecerá sus propios valores finales de posición, los cuales se compararán con los generados con la presente *toolbox*, para así obtener el error o variación entre aquel lugar que se quiere alcanzar y el verdaderamente alcanzado.

3 DESARROLLO DEL ESTUDIO

En este capítulo se procederá a describir las actividades desarrolladas a lo largo del proyecto, explicando los inconvenientes encontrados durante el desarrollo del mismo y describiendo las soluciones adoptadas. Para ello se dividirá el capítulo en distintos apartados asociados a cada etapa del diseño. Estas etapas se describen en el siguiente apartado.

3.1 DESCRIPCIÓN GENERAL

En este primer apartado se busca dotar al lector de una visión general del proceso que se pretende llevar a cabo con el fin de obtener el entorno de simulación virtual objeto de este proyecto. Con dicha finalidad se muestra el esquema de la figura siguiente:

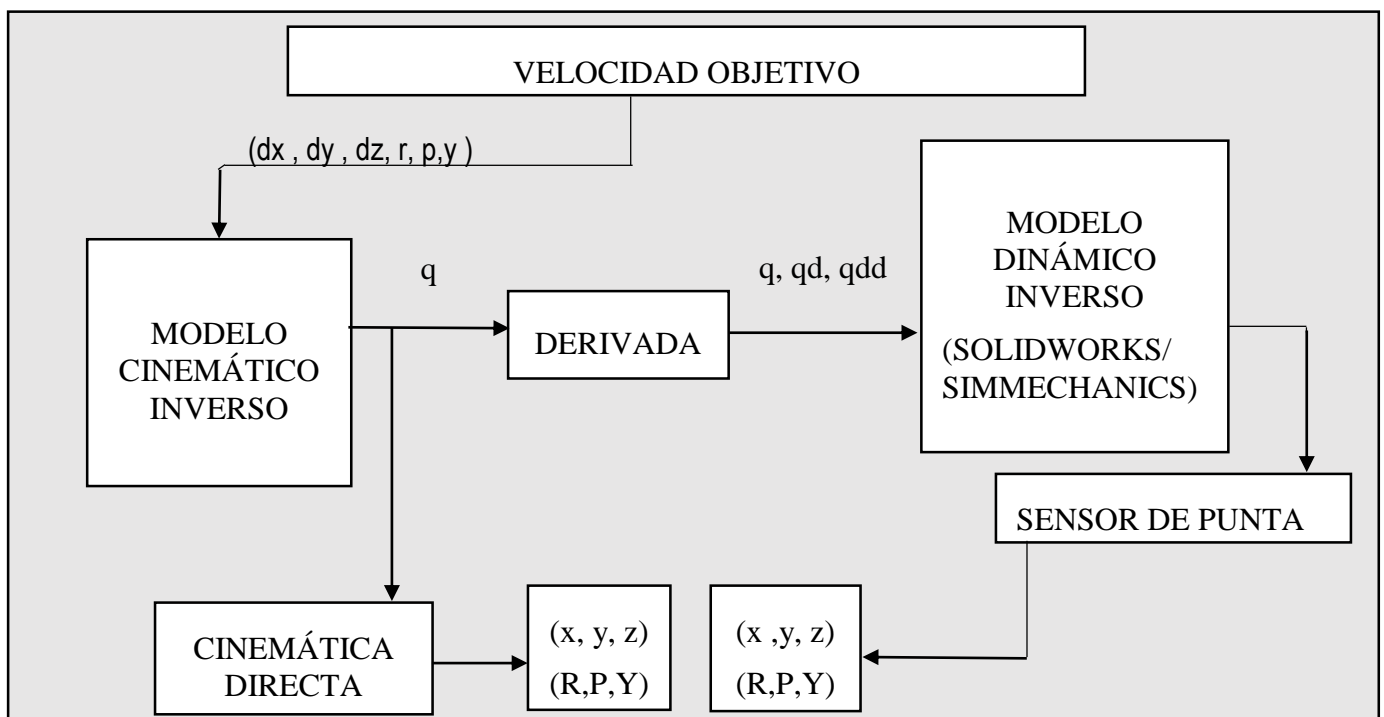


Figura 8 Esquema descriptivo del Entorno de Simulación

3.2 BLOQUE CINEMÁTICO INVERSO

Como se puede observar en el esquema anterior, se partirá de unos valores cartesianos de posición o velocidad deseada, identificados en la figura 8 como POSICIÓN/VELOCIDAD OBJETIVO. Estos parámetros iniciales son usados como entrada para el bloque CINEMÁTICO INVERSO.

La cinemática inversa es la técnica mediante la que se determina el movimiento a realizar por el conjunto o cadena de articulaciones (*joints*) que conforman un actuador con el fin de ubicarlo en la posición concreta. Para ello la cinemática inversa se encarga de encontrar las coordenadas de cada articulación del robot para posicionar su extremo en base a la localización espacial deseada. Sin embargo, esta técnica es un problema complejo, ya que consiste en la resolución de un conjunto de ecuaciones y que además puede no tener una solución única. Existen numerosos métodos resolutivos, como pueden ser:

- Métodos geométricos.
- Resolución a partir de matrices de transformación homogénea.
- Desacoplo cinemático.
- Cuaterniones duales.
- Métodos iterativos.

Como consecuencia de esta complejidad, se ha decidido resolver el problema de la cinemática inversa afrontándolo desde otra perspectiva que será explicada en los siguientes apartados.

De cualquier modo, la primera actividad a realizar es definir el actuador objeto de estudio. Para ello se ha utilizado el método de Denavit-Hartenberg consistente en una técnica matricial con la que se establece de manera sistemática un sistema de coordenadas ligado a cada eslabón de una cadena de articulaciones, pudiendo determinar así las ecuaciones cinemáticas de la cadena completa del brazo antropomórfico que compone el MANFRED 3, denominado LWR 2. Por tanto, teniendo los parámetros DH es posible definir el robot articulado. Estos parámetros fueron proporcionados dentro de la documentación de entrada en base a la que realizar este estudio y se muestra a continuación:

Tabla 2 Parámetros DH

j(joints)	theta (joint angle)	d(link offset)	a (link length)	alpha(link twist)	Offset(joint variable offset)	sigma
1	q1	0.228	0	-1.571	0	Revolute
2	q2	0	0	1.571	0	Revolute
3	q3	0.3125	0	-1.571	0	Revolute
4	q4	0	0	1.571	0	Revolute
5	q5	0.278	0	-1.571	0	Revolute
6	q6	0	0	1.571	0	Revolute
7	q7	0.18	0	0	0	Revolute

Conocidos estos parámetros se procede a definir el manipulador. Para ello se utilizará la herramienta software de Peter Corke.

3.2.1 HERRAMIENTA PETER CORKE

La *toolbox* de Peter Corke permite generar modelos robots y sus simulaciones. Mediante el uso de esta herramienta se obtendrá el esquema de bloques de Simulink que conformarán el modelo cinemático inverso. El conjunto de actividades a realizar mediante esta *toolbox* pueden dividirse en:

- Definición del brazo antropomórfico LWR 2.
- Generar las funciones con las que resolver la cinemática inversa.
- Obtención de los valores articulares de posición (q), velocidad (\dot{q}) y aceleración (\ddot{q}).

3.2.1.1 Definición del brazo LWR 2

La primera tarea a realizar es definir el brazo antropomórfico. Para ello es necesario familiarizarse con la sintaxis de programación que utiliza la *toolbox* y que viene recogida en el libro asociado a esta herramienta SW, [7].

El manipulador objeto de este análisis está conformado por una base y un brazo robótico articulado. Este brazo se compone de un conjunto de cuerpos (*links*) los cuales están unidos en serie. La unión entre dos links se realiza mediante una articulación o joint. De este modo la estructura articulada del robot puede ser descrita como una cadena. Para un manipulador con N articulaciones enumeradas de 1 a N , habrá así $1+N$ *links*, enumerados desde 0 a N , siendo el link 0 la base del manipulador y siendo N el link final del que se obtendrá la posición final del manipulador.

La *toolbox* ofrece un sistema sencillo para definir un manipulador y que se ejemplifica en la siguiente imagen, obtenida del libro, [7] y donde, en base a los parámetros DH de referencia se define un robot de dos links:

Link	θ_i	d_i	a_i	α_i	σ_i
1	q_1	0	1	0	0
2	q_2	0	1	0	0

```
>> L(1) = Link([0 0 1 0]);
>> L(2) = Link([0 0 1 0]);
>> L
L =
    theta=q1, d=0, a=1, alpha=0 (R,stdDH)
    theta=q2, d=0, a=1, alpha=0 (R,stdDH)
```

Figura 9 Definición robot 2-links con el manipulador, [7]

A partir de este ejemplo ilustrativo se puede proceder a definir el MANFRED. El código para generar el manipulador puede encontrarse en el ANEXO II. Definido el brazo, se obtiene su esquema gráfico

mediante el comando *plot*, a través del que se puede comprobar si la definición del brazo ha sido la correcta:

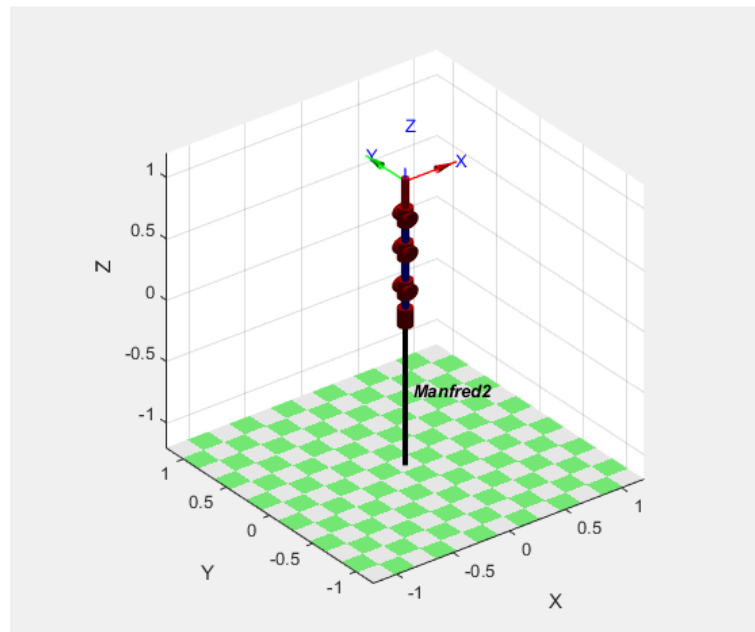


Figura 10 Representación gráfica del manipulador

Además de las distintas opciones generales de este tipo de representación gráfica, como son cambiar la vista de la estructura generada, girarla, ampliarla, etc., una de las acciones más interesantes es que es posible modificar la posición de cada articulación del brazo antropomórfico dentro de este entorno gráfico, lo que permite alcanzar de un modo visual y más intuitivo la posición que se desee. Esto se consigue a través del comando *teach*, que genera un cuadro desde el que modificar cada articulación de acuerdo con lo mostrado en la figura siguiente:

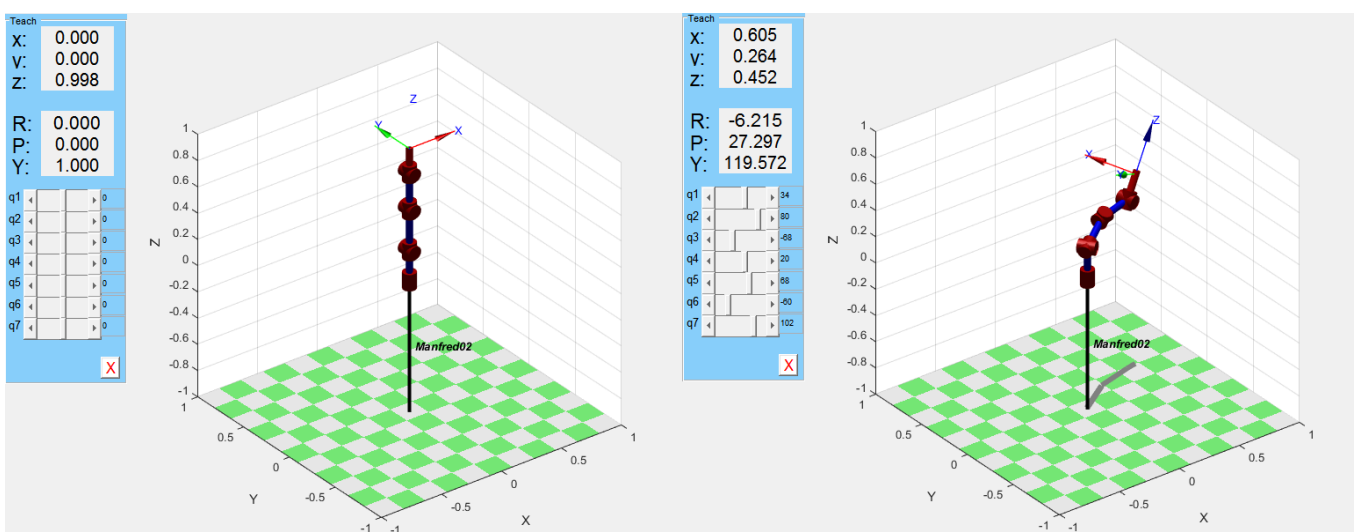


Figura 11 Modificación Posición del manipulador

Definido el manipulador es útil ejecutar otra de las soluciones que ofrece la *toolbox*. La herramienta permite generar los bloques de Simulink asociados a los parámetros del brazo antropomórfico que se ha generado.

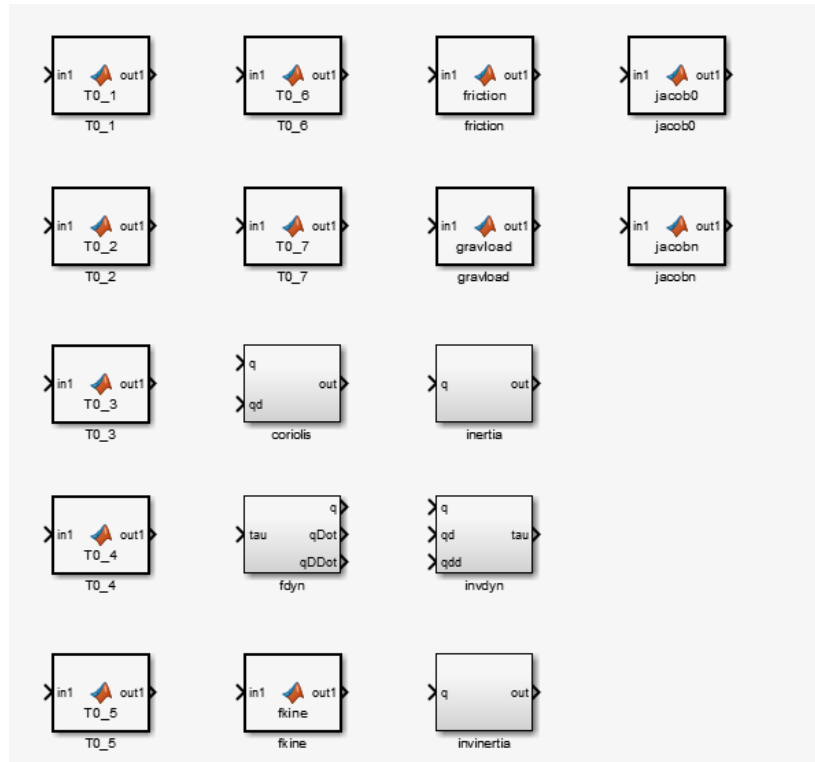


Figura 12 Bloques asociados al brazo

Estos bloques mostrados en la Figura 12 serán, por tanto, de gran utilidad a la hora de generar el entorno de simulación en Simulink. Definido el brazo se procede a dar solución al problema cinemático inverso.

3.2.1.2 Solución al problema cinemático inverso

Retomando la explicación descrita en el inicio del apartado 3.2, la cinemática inversa se encarga de encontrar las coordenadas articulares de cada articulación del robot para posicionar su extremo en base a la localización espacial deseada. Conocidas dichas coordenadas articulares puede entonces simularse el movimiento del brazo.

Como se indicaba, la resolución de la cinemática inversa es un problema complejo dado que soluciones distintas entre sí pueden ser igualmente válidas, por lo que existen numerosos métodos resolutivos.

A continuación se describen los métodos resolutivos barajados para solucionar el problema cinemático en el presente estudio, indicando igualmente la solución final adoptada.

○ Método 1: TRABAJO CON POSICIONES CARTESIANAS:MÉTODO ITERATIVO

La primera opción para generar el movimiento de la simulación está basada en el movimiento en el desplazamiento en el espacio cartesiano(X, Y, Z).

Considérense dos posiciones cartesianas distintas T1 y T2, por ejemplo:

T1= (0.4, 0.2, 0)

T2= (0.4, -0.2, 0)

En base a los valores de posición en coordenadas cartesianas se pueden calcular sus coordenadas articulares asociadas mediante el comando de la *toolbox ikine*, lo que calcula directamente la cinemática inversa o, lo que es lo mismo, las coordenadas articulares asociadas a una posición cartesiana dada.

Una vez conocidos los valores articulares (q1 y q2 respectivamente) se define el tiempo (t) en que se desea que se produzca este desplazamiento.

Con todos estos datos, la herramienta SW calcula las velocidades (qd) y aceleraciones articulares (qdd) asociadas al desplazamiento en función del tiempo dado (t) mediante una función preestablecida por la propia herramienta denominada *jtraj*:

```
>>[q , qd, qdd]= jtraj (q1, q2, t)
```

Obtenidos todos los valores, la simulación puede generarse gráficamente a través el comando *plot* explicado en apartados anteriores. Además, ya se dispone así de todos los valores necesarios para insertar en el diagrama de SimMechanics.

○ Método 2: TRABAJO CON VELOCIDADES CARTESIANAS: JACOBIANA

La segunda opción toma como base de entrada la velocidad a la que se desea que se mueva la punta del manipulador. Conocidas las velocidades cartesianas se pueden obtener las velocidades de los motores o velocidades de cada articulación.

Si en el método anterior se utilizaba la cinemática inversa para obtener la conversión, esta opción está basada en la matriz jacobiana inversa, que a partir de los valores de velocidad del extremo del robot determina las velocidades articulares (fundamentos teóricos en [7]).

La siguiente ecuación (1) muestra el procedimiento sobre cómo obtener esta conversión:

$$\dot{q} = J(q)^{-1} v \quad (1)$$

Donde:

\dot{q} = velocidad articular

$J(q)^{-1}$ = Jacobiana inversa de la posición inicial q

v = Velocidad deseada (m/s)

Obtenidas las velocidades articulares la función JTRAJ descrita en el método 1 también es útil en este segundo método ya que de un modo similar al descrito anteriormente también se obtienen las aceleraciones articulares, teniendo así todos los valores a utilizar como entrada en el robot en SimMechanics.

○ OPCIÓN ESCOGIDA

Una vez expuestos los dos métodos posibles a utilizar es necesario decidir cuál de ellos escoger.

Dado que la finalidad de este proyecto es la de realizar una simulación dinámica inversa en SimMechanics del manipulador, tiene más sentido que la trayectoria final venga dada en velocidades, ya que es un valor que se necesita controlar y así asegurar la velocidad con la que se realizan los movimientos. Este control de la velocidad no es posible si se hace uso de la cinemática inversa. Por esto se ha elegido el método 2 para la realización de la simulación.

3.2.1.3 DISEÑO DE LA SIMULACIÓN

Elegido el método de simulación del robot, a continuación se procederá a explicar su diseño de un modo más detallado. A modo de explicación, se va a buscar realizar un esquema similar al siguiente:

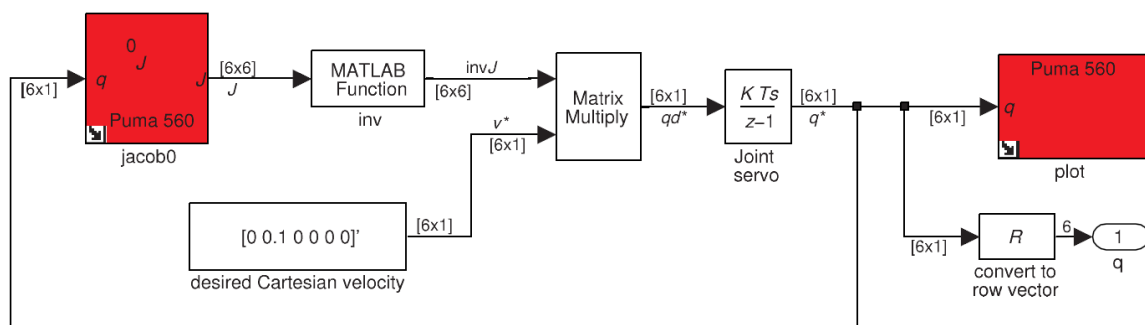


Figura 13 Esquema cálculo movimiento-Jacobiana explicativo, [7]

Retomando la ecuación (1) del apartado anterior y el esquema de la Figura 13, se partirá de la velocidad a la que se desea que se mueva el extremo del manipulador para calcular el resto de parámetros y así

conectarlo al bloque dinámico inverso. Para realizar este procedimiento se necesitarán conocer los siguientes datos iniciales:

q_0 = Coordenadas articulares iniciales del manipulador

v = velocidad de desplazamiento deseada

Como se puede observar ambos datos son conocidos ya que los definimos acorde a nuestras preferencias.

A su vez el bloque Jacobiana asociado al brazo robótico ya se generó en el apartado 3.2.1.1. (Figura 12).

Teóricamente ya se dispone de todos los bloques necesarios para generar un esquema similar al de la figura anterior. Sin embargo es necesario realizar un paso más que se explica a continuación.

El brazo antropomórfico objeto de este estudio tiene 7 GDL. Esto hace que se trate de un robot conocido como redundante o también por su término en inglés: “*over-actuated robot*”, ya que tiene más de 6 GDL. Esto provoca que la matriz Jacobiana sea más ancha que alta, o lo que es lo mismo, tener una matriz de más columnas que filas.

Para los robots redundantes, la ecuación de obtención de la velocidad articular en función de su Jacobiana sufre ligeros cambios, de acuerdo con lo descrito en el manual de robótica, [7] y que se describe en la ecuación 2:

$$\dot{q} = J(q)^+ v \quad (2)$$

Como puede observarse la Jacobiana ha cambiado su signo. Este signo (+) lo que indica es que en lugar de realizar la inversa Jacobiana, para los robots redundantes hay que realizar la pseudoinversa. Gracias a la pseudoinversa, del conjunto de soluciones posibles, se producirá aquel para el cual $|\dot{q}|$ es el más pequeño.

Conocido este cambio, se puede generar el esquema equivalente a la ecuación 2:

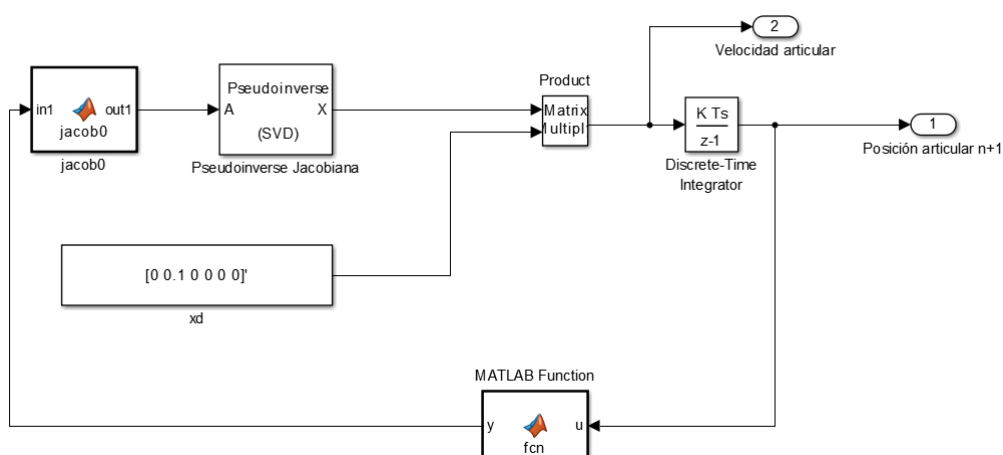


Figura 14 Esquema cálculo velocidad articular

Donde:

-*jacob*= bloque de cálculo de la jacobiana

-*Pseudoinverse Jacobiana*: bloque cálculo de la pseudoinversa

-*Matrix multiply*: bloque multiplicador de la velocidad deseada y la salida de la pseudoinversa

- \dot{x}_d = velocidad cartesiana deseada

-*Discrete time integrator*: Con este bloque se integra la velocidad articular. De este modo se obtiene la siguiente posición articular del manipulador en función de la velocidad deseada. Esta siguiente posición articular q_{n+1} se realimenta al bloque Jacobiana. El bloque de realimentación intermedio solo tiene como finalidad invertir el vector posición de salida, pasando de ser $[7 \times 1]$ a $[1 \times 7]$. Además, dentro de este bloque se puede fijar la posición inicial articular del brazo antropomórfico q_0 deseada, acorde a la figura siguiente:

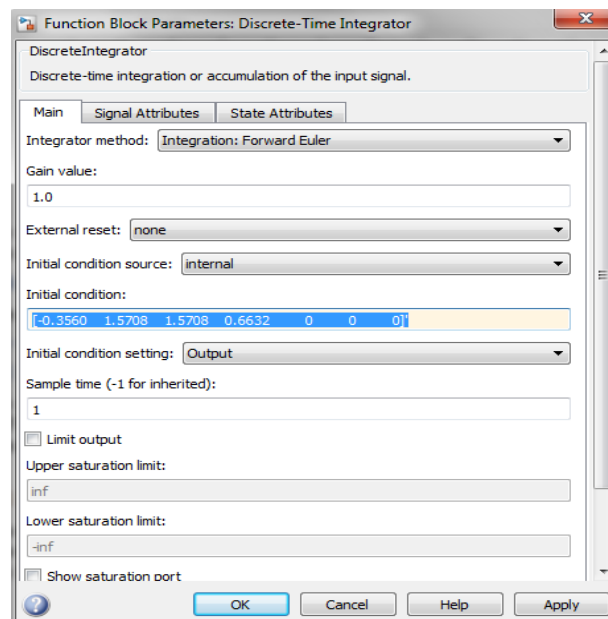


Figura 15 Discrete-Integrator: Fijado de posición inicial

Como se definía en la Figura 14 , del esquema se obtienen dos parámetros de salida:

- Salida 1: Posición articular $n+1$: Vector $[7 \times 1]$, que contiene los siete valores de posición articulares asociados a cada una de las articulaciones del brazo.
- Salida 2: Velocidad articular: Vector $[7 \times 1]$ que contiene los siete valores de velocidad asociados a cada una de las articulaciones del brazo.

Estos valores, además de los parámetros de aceleración asociados a cada articulación, son los valores de entrada a utilizar en el bloque dinámico inverso. Conocido esto, las cuestiones que se plantean en este punto son las siguientes:

- Obtención de los valores de aceleración asociados a cada articulación.
- Conocer las posiciones cartesianas asociadas en cada momento.

3.2.1.3.1 Obtención de los valores de aceleración articulares

Para solventar la cuestión de la obtención de los valores articulares de aceleración la primera solución que se nos puede ocurrir es la de utilizar un bloque derivada y así, del mismo modo que para obtener el valor de posición se utilizó el bloque integrador (*discrete integrator*), para obtener los valores de

aceleración se puede incluir un bloque (denominado *Derivate*) que calcule la derivada de la salida del esquema de la Figura 14.

El segundo método es utilizar los bloques de funciones facilitados por la *toolbox* de Peter Corke donde se ofrece una función con la que poder obtener los parámetros requeridos.

Esta función, denominada *jtraj*, genera los parámetros de aceleración en base a un conjunto de parámetros o valores de entrada. Este conjunto de parámetros pueden observarse en la siguiente figura:

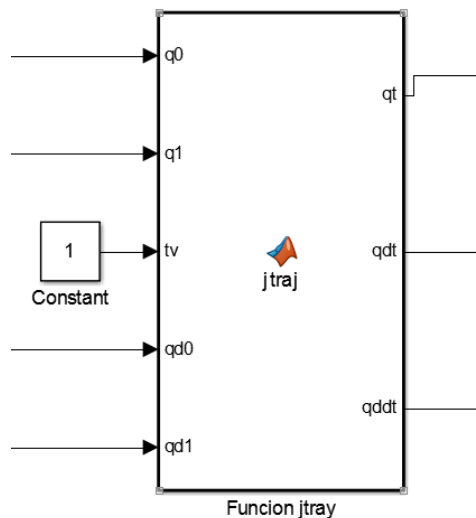


Figura 16 Bloque jtraj

Donde:

q_0 : Posición articular inicial n

q_1 : Posición articular $n+1$

tv : Número de pasos en los que alcanzar siguiente posición

qd_0 : Velocidad articular inicial n

qd_1 : Velocidad articular $n+1$

Todos los valores son conocidos. Tanto q_0 , q_1 , qd_0 y qd_1 se obtienen del esquema Jacobiana, Figura 14, mientras que el valor tv es un valor que se define según el número de pasos en los que se quiera alcanzar la siguiente posición, como se indicaba en el texto de la figura. En este proyecto ese valor será 1 ya no es necesario subdividir el número de pasos a realizar para alcanzar la siguiente posición. Por tanto es posible aplicar este bloque directamente obteniendo como salida los valores articulares de posición, velocidad y aceleración a servir como entrada en los distintos bloques de SimMechanics. El esquema obtenido se muestra a continuación:

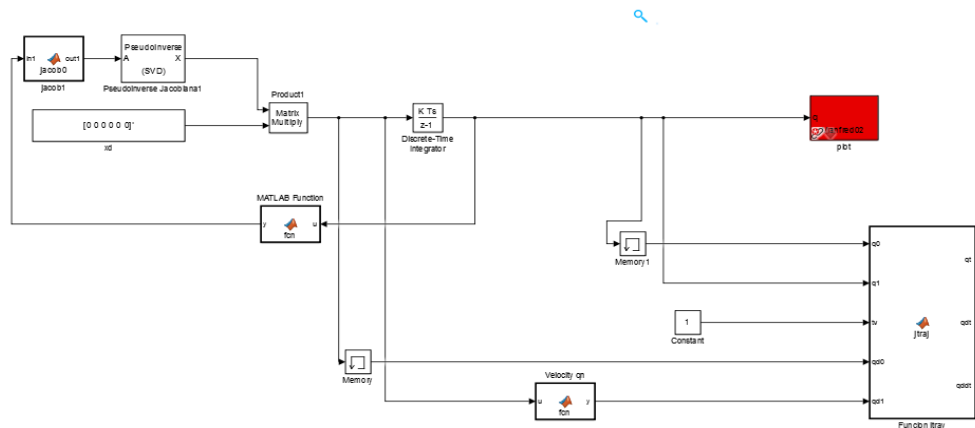


Figura 17 Esquema cinemático inverso mediante jtraj

Como se puede observar, además del esquema de Jacobiana y la adición del bloque *jtraj* existe además un bloque rojo denominado *plot*. Este bloque, del mismo modo que ocurría con la función *plot* descrita anteriormente, se encargará de generar la simulación gráfica de la *toolbox* de Peter Corke.

Aunque inicialmente este fue el esquema utilizado, durante las simulaciones se encontraron numerosos errores asociados al bloque *jtraj*. Debido a este hecho finalmente se decidió suprimir este bloque y hacer uso de la primera opción barajada para la obtención de los parámetros de aceleración articulares. Mediante los bloques de derivada, se obtienen los valores articulares de velocidad y aceleración asociados a una posición articular determinada. De este modo el esquema queda según lo mostrado en la siguiente figura.

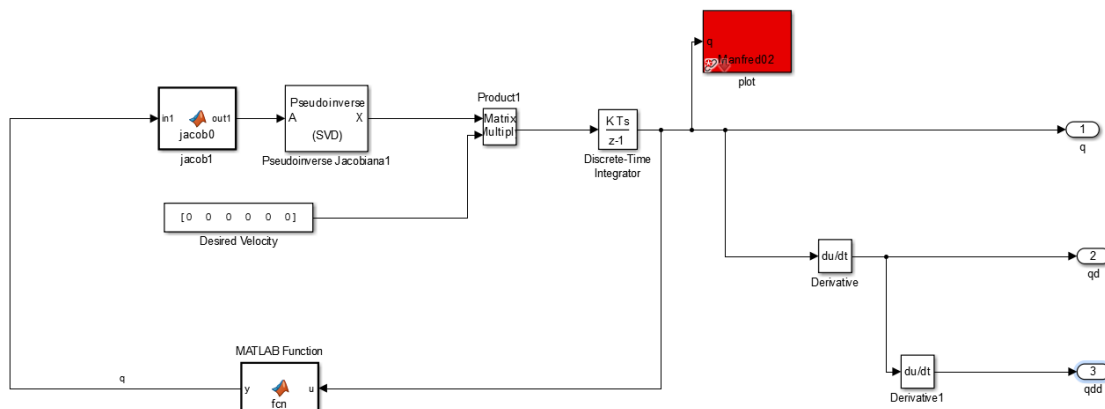


Figura 18 Modelo cinemático inverso final

Con esta configuración queda definido el esquema cinemático inverso y se puede proceder a realizar la conexión al bloque dinámico inverso. Sin embargo, antes de dar este paso también interesa conocer los valores cartesianos equivalentes a los parámetros articulares que se están usando como entrada en el bloque dinámico inverso. De este modo, en base a los valores de posición articulares conocidos se calcula la cinemática directa del manipulador. La cinemática directa calcula la posición del extremo o final de un sistema articulado a partir de sus valores articulares. Para el caso de estudio que este proyecto ocupa, para calcular la cinemática directa (FK) se utilizará uno de los bloques que se generaron

en la Figura 12 denominado *fkine*. Este bloque tendrá como entrada la posición q y como salida la matriz de transformación homogénea T asociada a dicha posición articular.

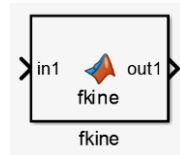


Figura 19 Bloque Cinemática Directa del brazo

Una matriz de transformación homogénea T es una matriz con dimensiones $[4 \times 4]$ que representa la transformación de un vector de coordenadas homogéneas de un sistema de coordenadas a otro. A su vez, esta matriz T se puede considerar que se compone de cuatro submatrices según se muestra en la siguiente ecuación (4):

$$T(4 \times 4) = \begin{bmatrix} R(3 \times 3) & p(3 \times 1) \\ f(1 \times 3) & W(1 \times 1) \end{bmatrix} = \begin{bmatrix} \text{Rotación} & \text{Traslación} \\ \text{Perspectiva} & \text{Escala} \end{bmatrix} \quad (3)$$

Con estos fundamentos se pueden obtener los valores correspondientes a la posición y orientación del extremo del manipulador. Para ello se utilizarán los bloques existentes en la librería de la *toolbox*. Estos bloques se muestran a continuación:

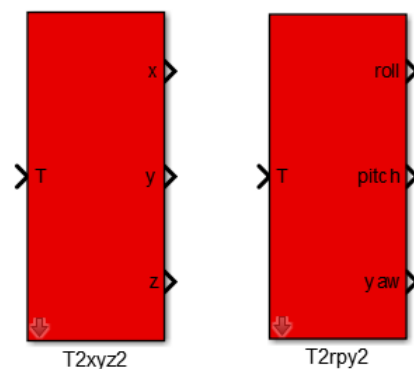


Figura 20 Bloques librería *toolbox* para transformada homogénea

Como se puede observar ambos bloques tienen como entrada la transformada homogénea T . Mientras que el bloque de la izquierda da como salida los valores X, Y, Z (posición cartesiana), el bloque de la derecha proporcionará los valores angulares del extremo articular del robot. De este modo, el esquema será el siguiente:

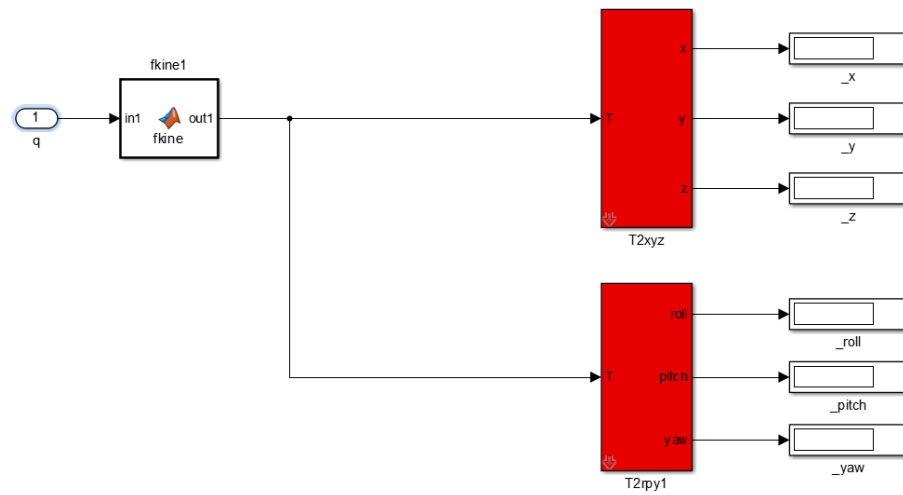


Figura 21 Cinemática Directa

Con la inclusión de estos dos bloques se puede dar por finalizado el apartado de modelado con la *toolbox* de Peter Corke.

3.3 BLOQUE DINÁMICO INVERSO

El segundo bloque del entorno de simulación a generar lo comprende la obtención del modelo dinámico inverso del MANFRED 3 (LWR2 + Plataforma). Con este modelo dinámico se pretende simular el movimiento del manipulador a partir de los valores articulares calculados en el apartado anterior. De este modo, a través del modelo dinámico inverso podrán obtenerse los valores de posición cartesianos del robot. La obtención del modelo dinámico puede dividirse en dos apartados:

- Modelado en SolidWorks del manipulador.
- Modelado en SimMechanics.

3.3.1 SOLIDWORKS

Como se indicaba en el capítulo 2.1 de este estudio, se ha utilizado la herramienta SW CAD SolidWorks 2014 para diseñar el manipulador robótico MANFRED 3. Con esta herramienta se modelan las piezas que constituyen el manipulador, permitiendo caracterizarlas con las propiedades que tendría esa pieza en la realidad como son su masa, centro de masa o momento de inercia.

Previo a la realización de este estudio ya existía un diseño del manipulador, como puede observarse en la Figura 22:

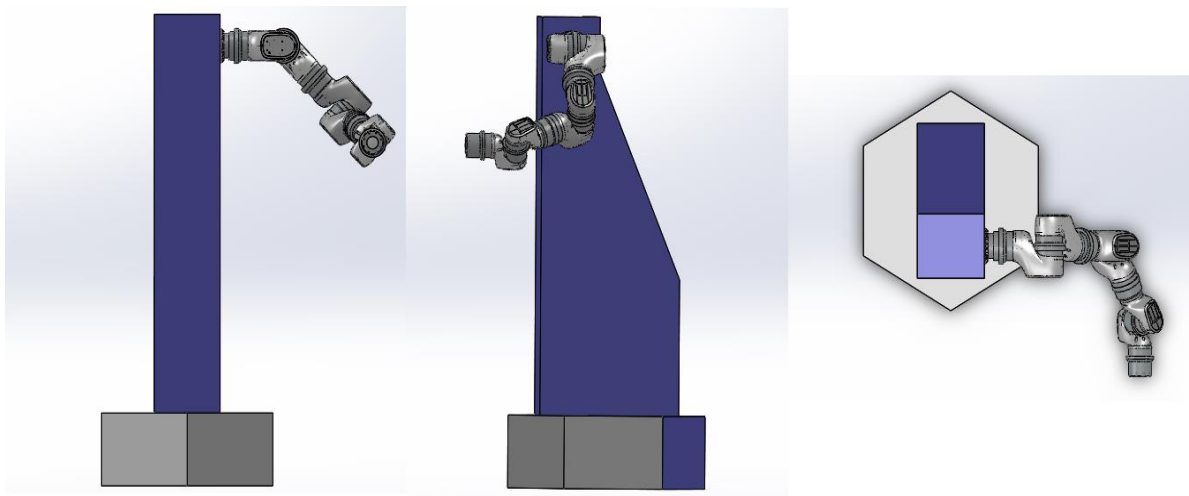


Figura 22 Situación inicial manipulador MANFRED 3

Sin embargo, por cuestiones de diseño, el brazo debe ir en el lado contrario de la plataforma. Por ello la primera fase de este diseño ha consistido en la reconstrucción del manipulador, volviendo a unir el conjunto de elementos que lo componen y reubicándolo en el lado correcto. Para este fin se ha utilizado el diseño de las distintas piezas que ya existía y que se muestran en el Anexo III de este proyecto.

El conjunto de pasos requeridos para la reconstrucción del manipulador se describe a continuación.

3.3.1.1 Paso 1: Nueva plataforma

El primer paso ha sido generar la estructura o plataforma base con los taladros que sirven de unión al robot en el lado correcto. Existen varias opciones para realizar este taladrado. En este caso, dado que

ya existía una estructura taladrada, la solución adoptada ha sido recurrir a una opción que ofrece la herramienta denominada simetría con la que creamos una nueva estructura simétrica a la existente como puede observarse en la Figura 23.

Cabe destacar que es necesario prestar especial atención al hacer uso de esta opción que ofrece el programa ya que, aunque inicialmente es un procedimiento sencillo, hay que ser muy cuidadosos con el modo de realizar la simetría (respecto a una cara, a un plano...), pues, aunque la estructura estéticamente no sufre cambios los ejes de ésta pueden verse alterados, hecho que no genera problemas a la hora de ensamblar la pieza, pero sí en el momento de realizar la simulación. Durante la realización del presente estudio se produjo este imprevisto, por lo que fue necesario volver a generar el ensamblaje con la plataforma correcta.

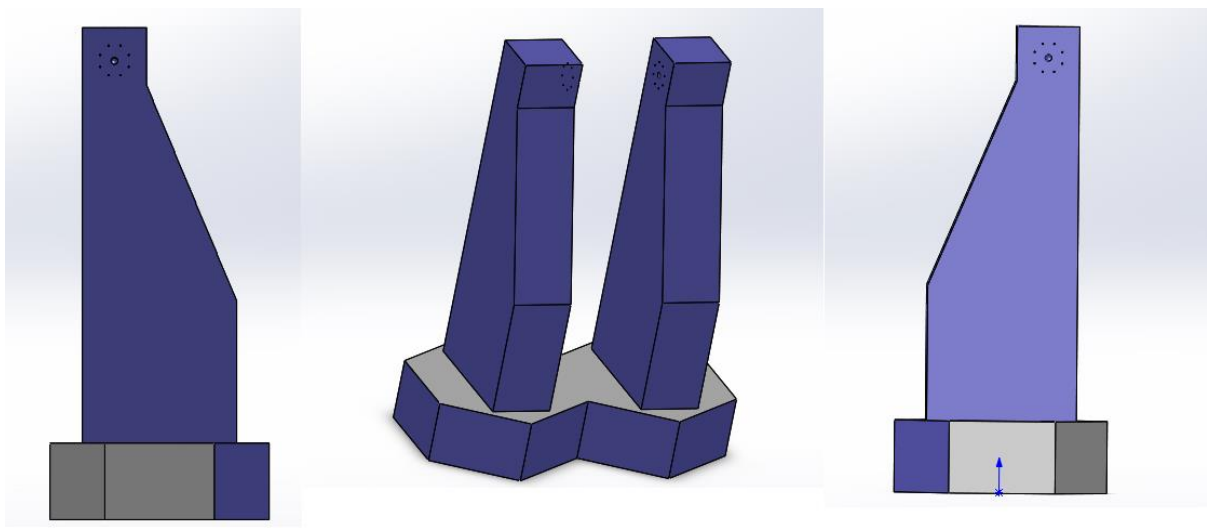


Figura 23 Evolución estructura. Estado inicial, simetría y estado final

3.3.1.2 Paso 2: Reconstrucción de los elementos

La reconstrucción ha consistido en volver a unir el conjunto de elementos que conforman el manipulador. Para este fin ha sido muy útil la opción de relaciones de posición que ofrece la herramienta, base de todo el proceso constructivo. Para su montaje se ha seguido como referencia el documento del diseño, [1] y el archivo de SolidWorks con el montaje en el lado contrario mostrado en la Figura 22 ya que, como se comentó anteriormente, aunque el robot estaba soldado en el lado equivocado, la unión entre los distintos elementos era la correcta.

De este modo, el esquema final queda tal y como se muestra en la siguiente figura (Figura 24)

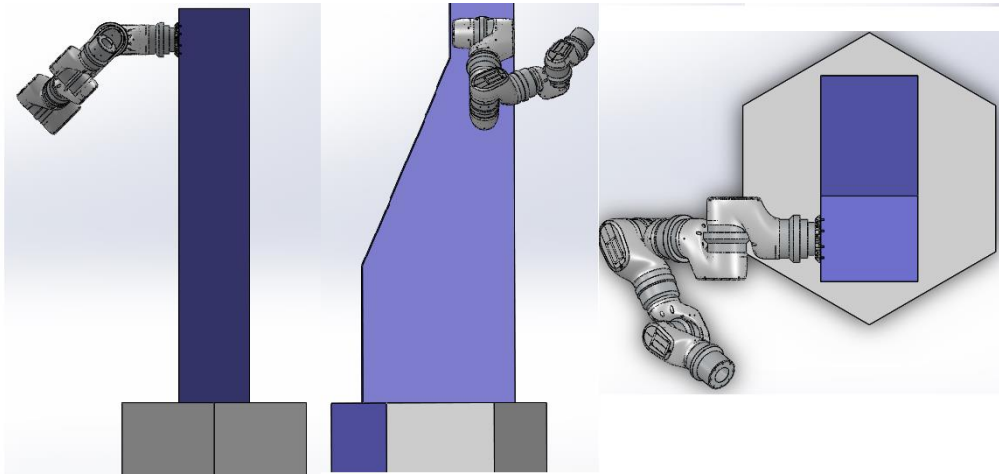


Figura 24 MANFRED 3, situación final con tapas

3.3.2 SIMMECHANICS

Una vez reconstruido el manipulador, el siguiente paso es exportar este diseño a Matlab, herramienta software en la que se va a crear el entorno de simulación. Para ello se ha utilizado el complemento de Matlab denominado SimMechanics Link, que ofrece una función agregada al entorno de la herramienta SolidWorks, mediante el cual exportar el archivo de diseño en formato .xml, que es el formato de los archivos de Matlab, generando de esta manera un diagrama de bloques del brazo. En este punto es importante destacar que resulta de vital importancia la posición en la que se encuentra el manipulador en el entorno de SolidWorks en el momento de realizar la simulación. Esto se debe a que SimMechanics hereda dicha posición como su posición inicial. Por esto, exportar la pieza en una posición óptima para el estudio que se lleva a cabo es muy importante ya que con esto se evitan numerosos errores de ajustes que se pueden producir en la simulación.

A modo resumen, a continuación se describen algunos de los bloques que conforman un esquema genérico de SimMechanics, el cual está conformado por un conjunto de bloques de distintas características según lo que éstos busquen representar y se unen entre sí mediante señales. Algunos de estos bloques se explican a continuación:

ROOTGROUND: Eje de ordenadas que utiliza el sistema de referencia.

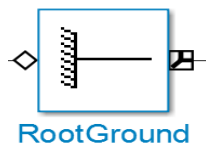


Figura 25 RootGround

WELD: Utilizado para indicar la unión/soldadura entre dos sólidos.

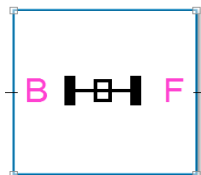
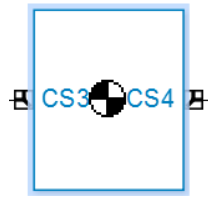
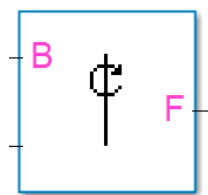


Figura 26 Weld

BODY: Representa un sólido.

Figura 27 *Body*

REVOLUTE: Representa grado de libertad rotacional, es decir, la relación de rotación entre dos sólidos.

Figura 28 *Revolute*

Re-ensamblado el manipulador y generado su diagrama de bloques, se procede a realizar un análisis preliminar de este diagrama. Con este análisis visual se pretende modificar algunos bloques, bien sea reordenando e incluso eliminándolos, ya que el esquema que se genera a través del SW de SolidWorks puede contener algunos bloques que desde el punto de vista de SimMechanics son poco útiles o redundantes. Además, con esta modificación también se consigue una visión más rápida e intuitiva del conjunto.

Finalmente en base a la modificación descrita, el esquema queda según se muestra en la figura siguiente:

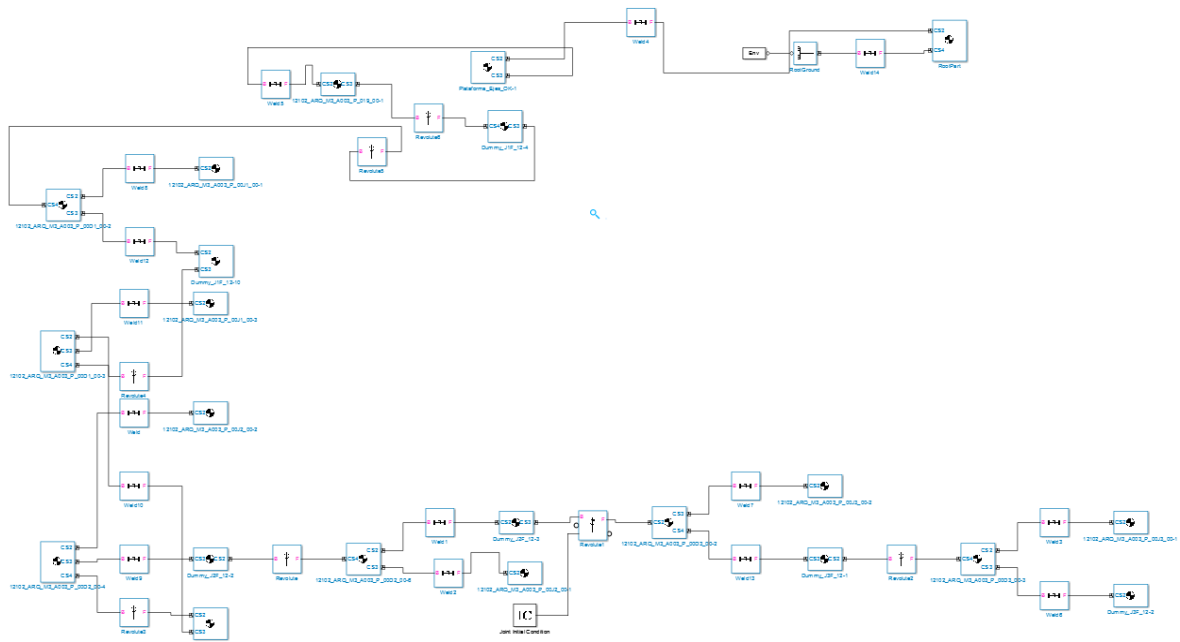


Figura 29 Esquema Simmechanics tras modificación inicial

Obtenido un diagrama más adecuado a los intereses del estudio se puede proceder a conectar los parámetros articulares de posición, velocidad y aceleración del bloque cinemático inverso al modelado dinámico inverso donde poder dotar de movimiento al brazo robótico que compone el manipulador y donde se tienen en cuenta el las inercias y pesos de cada eslabón en función de los materiales con los que el manipulador fue diseñado en SolidWorks que permiten obtener los pares asociados a los motores. Este proceso se describe en el siguiente apartado.

3.3.3 CONEXIONADO DE BLOQUES

Con los valores de posición, velocidad y aceleración articulares obtenidos en el bloque cinemático inverso Peter Corke se procede a utilizarlos como parámetros de entrada para el diagrama de bloques modelado en SimMechanics. El procedimiento será el siguiente:

Paso 1: Bloque *Demux*:

Dado que los parámetros de entrada del modelo cinemático son vectores de tamaño [1x7] es necesario convertir esta entrada en siete distintas, para así conectar cada uno de estos valores a su correspondiente bloque de SimMechanics.

Para realizar esta división se utiliza el bloque *Demux*, con el que se consiguen las siete entradas esperadas:



Figura 30 Bloque *Demux*

Deberán así utilizarse tantos bloques como entradas se tengan. Dado que para la realización de esta simulación se tienen entradas correspondientes a posición, velocidad y aceleración articulares, habrá entonces que utilizar tres bloques.

Paso 2: Conexión con el bloque Dinámico Inverso:

El conjunto de parámetros de entrada deben conectarse a cada uno de los bloques de revolución que se describían en la Figura 28. A cada uno de estos bloques hay que insertarle su parámetro correspondiente de posición, velocidad y aceleración articular. El procedimiento de conexión requerirá de un bloque *Mux*, que tomará las tres entradas y las unirá en una sola única señal. Esta señal a su vez se usará como entrada para el bloque denominado *Joint Actuator*, que simula un actuador. Este actuador será el que se conecte a cada bloque revolvente. El esquema completo se muestra en la figura siguiente:

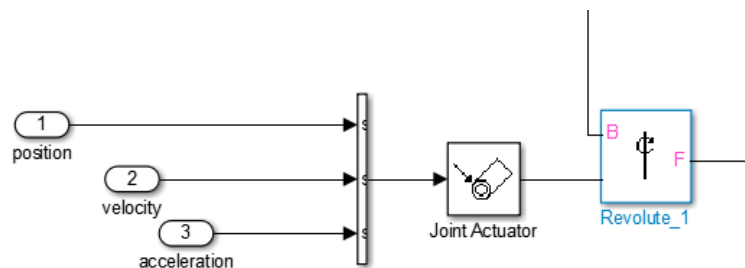


Figura 31 Conexión parámetros a SimMechanics

Este esquema deberá repetirse para cada uno de los bloques *revolute* de SimMechanics, que en total serán siete, que se corresponden con los siete valores de entrada de cada vector procedente del esquema de la herramienta de Peter Corke.

Una vez realizado todo el conexionado puede procederse a generar el análisis y ajuste de la simulación, con el fin de crear un entorno fiable y útil para futuros estudios. Para ello hay que realizar un último paso antes consistente en la inclusión de un sensor de posición en la punta o extremo del robot modelado en SimMechanics. Este sensor proporcionará los valores cartesianos alcanzados por el robot permitiendo así compararlos con los valores cartesianos que se obtuvieron mediante la cinemática directa descrita en el apartado 3.2.1. Esta comparación será la base que permitirá concluir si se está realizando o no una simulación acertada. El estudio de la simulación se desarrolla en el siguiente capítulo.

3.4 AJUSTES DEL ENTORNO DE SIMULACIÓN

Una vez completado el conexionado entre los bloques cinemático inverso y dinámico inverso es necesario realizar una serie de ajustes previos a proceder a realizar las pruebas de simulación que determinen la validez del entorno generado en este proyecto. Con estos ajustes se pretenden corregir posibles *offset* que existan entre ambos sistemas. Estos errores pueden deberse a distintos factores. Uno de los más importantes es el hecho de que estamos generando un brazo robótico a través de dos sistemas distintos de definición, el primero a través de la herramienta de Peter Corke y el segundo mediante SimMechanics. Además hay que agregar los posibles errores que SimMechanics pudiese heredar del modelado en SolidWorks, como pueden ser conexiones erróneas entre los elementos.

La primera diferencia se detecta simplemente visualizando los entornos gráficos de SimMechanics y de Peter Corke, como puede observarse en la siguiente figura:

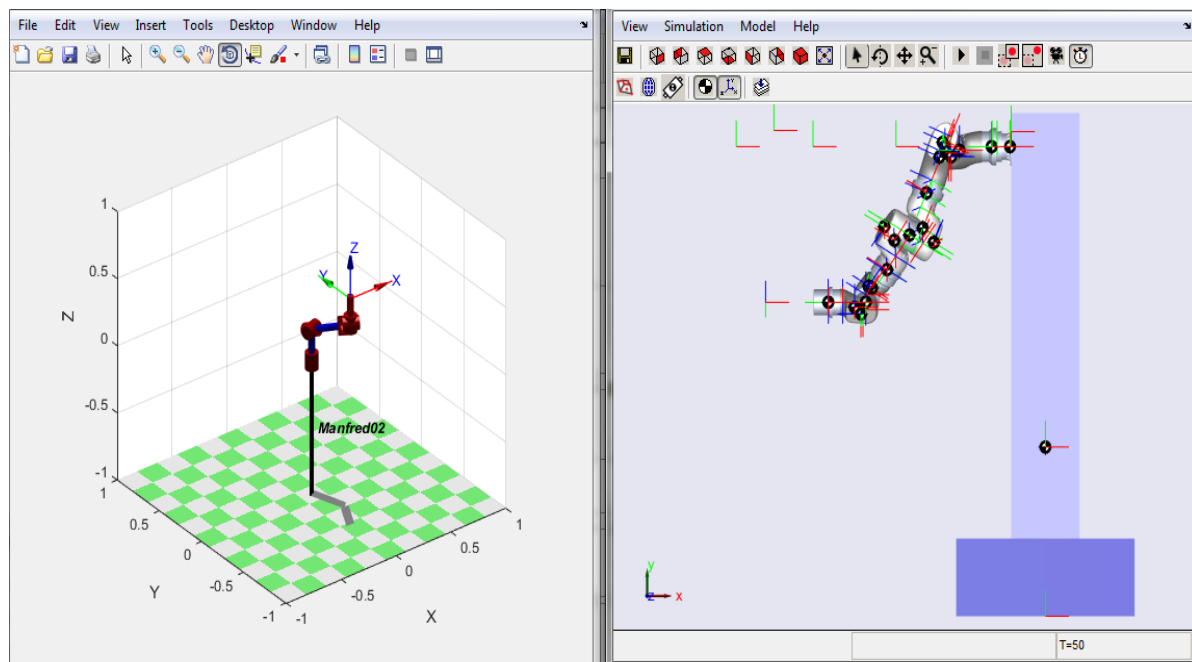


Figura 32 Entornos gráficos Peter Corke -SimMechanics

Analizando ambas imágenes se observa que el sistema de coordenadas de ambos entornos no es el mismo. Además, en el caso de SimMechanics se tiene una plataforma que no existe en Peter Corke y adicionalmente el origen de coordenadas se encuentra fuera del manipulador. Para corregir este problema se propusieron varias soluciones, como fueron:

- Modificar el sistema de coordenadas de SimMechanics mediante la modificación del modelo de SolidWorks. Se intentó cambiar el modelo de SolidWorks realizando distintas pruebas como fueron crear un sistema de coordenadas nuevo en el sólido, modificar el sistema de exportación a SimMechanics o crear un parasólido. Sin embargo todos los resultados obtenidos de estas pruebas fueron negativos por lo que fueron descartados.
- Calcular la matriz de rotación. Las matrices de rotación son un método matricial mediante el cual determinar vectores de giro sobre los ejes del sistema de coordenadas. De este modo teóricamente se consigue pasar de un sistema de coordenadas al deseado. Este método aunque es teóricamente posible, en la práctica no se produjo resultados satisfactorios.

En base a esto finalmente se optó por una solución más sencilla. Esta solución puede dividirse en dos. En primer lugar se ha incluido un *body sensor* en la pieza que da comienzo al brazo robótico en SimMechanics. Al ser una pieza soldada a la plataforma, su posición no variará a lo largo de la simulación. Restando esta posición a la posición de referencia del sistema completo se consigue tener un nuevo origen de referencia en el comienzo del brazo, siendo así coincidente con el inicio del sistema de coordenadas del entorno generado con la herramienta de Peter Corke. De este modo se elimina el problema de tener el origen fuera del manipulador.

Adicionalmente, analizando los ejes de ambos sistemas se obtiene la siguiente equivalencia:

$$X_{PC}=Z_{SM}$$

$$Y_{PC}=Y_{SM}$$

$$Z_{PC}=-X_{SM}$$

Tomando estas equivalencias como referencia, se corrige la salida del bloque dinámico inverso de acuerdo con el siguiente esquema:

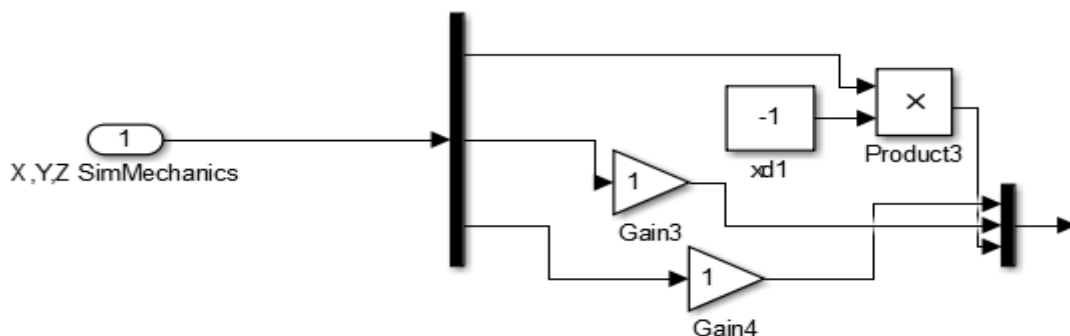


Ilustración 33 Ajuste salida bloque dinámico inverso (posición cartesiana extremo robot)

Cabe mencionar que con esta corrección se consigue es realizar una rotación de los datos de salida, corrigiendo así el problema de la posición de los ejes entre ambos sistemas, pero esta solución no corrige la diferencia de la orientación.

Estas correcciones, en cambio, seguían mostrando ciertas desviaciones entre los valores de salida del bloque dinámico inverso y los valores esperados, es decir, los valores cartesianos del bloque cinemático inverso. Fruto de este análisis se detectaron dos errores más. El primero de ellos ya se indicó en el apartado 3.2.1, correspondiente al bloque JTRAJ, Figura 16. Aunque inicialmente se intentó utilizar, las salidas de este bloque en ocasiones no eran coherentes. Por este motivo se decidió eliminar y hacer uso de los bloques derivativos.

El segundo error se encontró analizando visualmente el entorno de simulación de SimMechanics. Dicho error se puede observar en la siguiente figura:

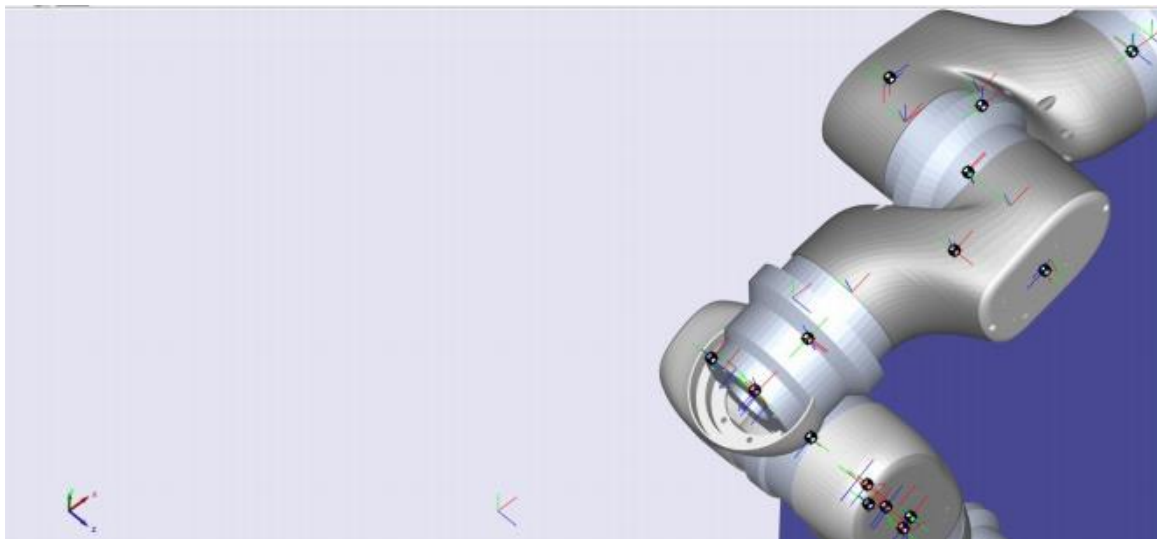


Ilustración 34 Error encontrado durante ajustes de simulación

Este error está asociado a las conexiones entre los bloques de SimMechanics. Aunque no se puede concluir con total rotundidad sobre el origen de este error, probablemente se produjo durante los ajustes visuales del esquema explicado en el apartado 3.3.2. Otro posible origen del error pudo producirse en el momento de exportar el modelado de SolidWorks. De cualquier modo, este error quedó corregido.

4 PRUEBAS DE SIMULACIÓN

En este capítulo se pretende demostrar si el entorno de simulación generado cumple las expectativas marcadas. Para ello se han definido un conjunto de movimientos y trayectorias a simular con las que comprobar si los datos de salida obtenidos concuerdan con los esperados, mostrando 4 gráficas por movimiento:

- En la primera de ellas se mostrarán las salidas del bloque cinemático inverso.
- La segunda corresponderá al bloque dinámico inverso, donde el manipulador está teniendo en cuenta también los parámetros asociados al peso e inercia.
- La tercera gráfica mostrará el error entre ambos sistemas.
- Por último se mostrará la trayectoria dibujada por el brazo.

Se debe señalar la importancia que tiene la correcta elección de velocidades y posiciones que se soliciten al robot en función de su posición inicial y de los movimientos que este brazo puede o no realizar.

4.1 PRUEBAS I: TRAYECTORIAS SIMPLES SOBRE CADA EJE

a- Movimiento sobre eje X a velocidad constante de 0.01m/s, Y=0m/s, Z= 0m/s

Salida del bloque cinemático inverso (m/s):

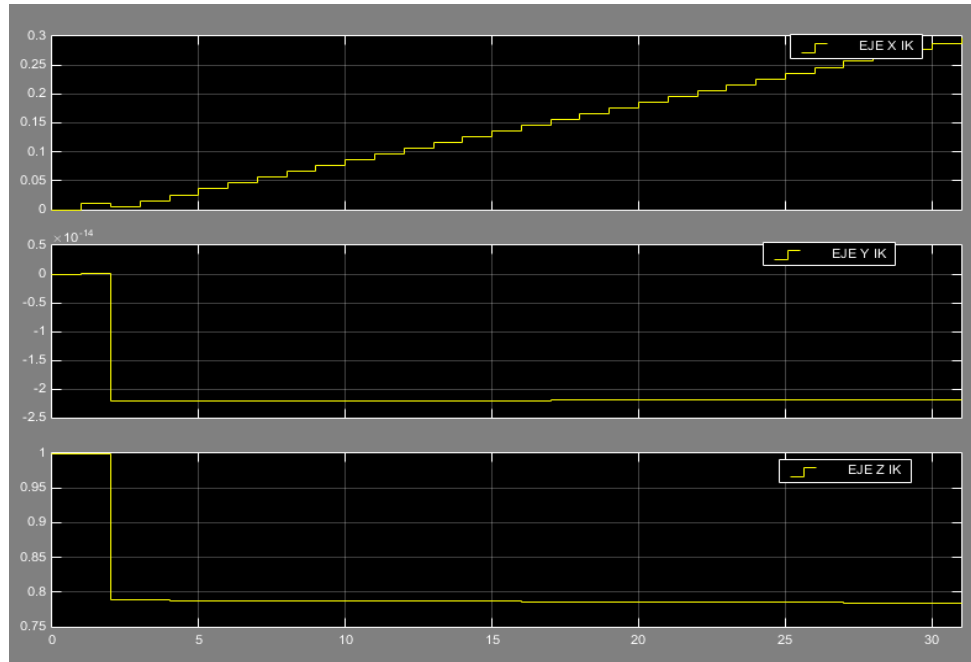


Figura 35 PI-A-Salida IK (m/s)

Salida del bloque dinámico inverso (m/s):

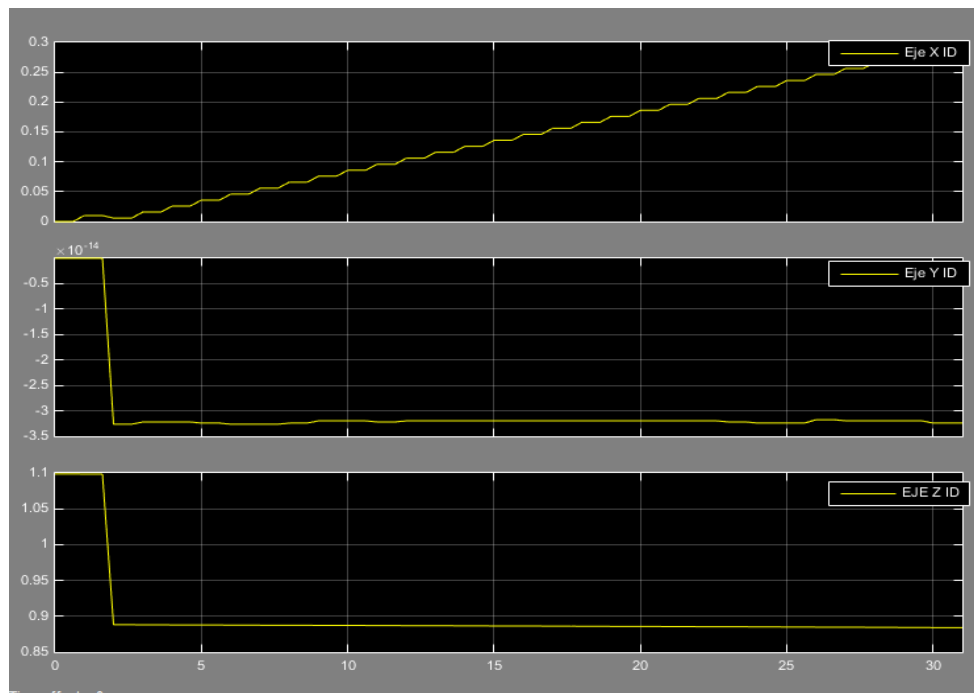


Figura 36 PI-A-Salida ID (m/s)

Tanto la salida del bloque ID como del bloque IK muestran un comportamiento similar. Durante los dos primeros segundos de la simulación se observa un salto o cambio brusco en los ejes Y y Z debido a la necesidad del brazo por adquirir una posición inicial que le permita realizar el movimiento. Una vez posicionado comienza a moverse a lo largo del eje X manteniendo a 0 los valores del movimiento en los ejes Y y Z, donde la velocidad de salida se ha marcado a 0 m/s.

A continuación se muestran los resultados de restar ambas señales, donde se observa que el error entre ambos sistemas es muy cercano a 0, indicador de que existe concordancia entre ambos sistemas

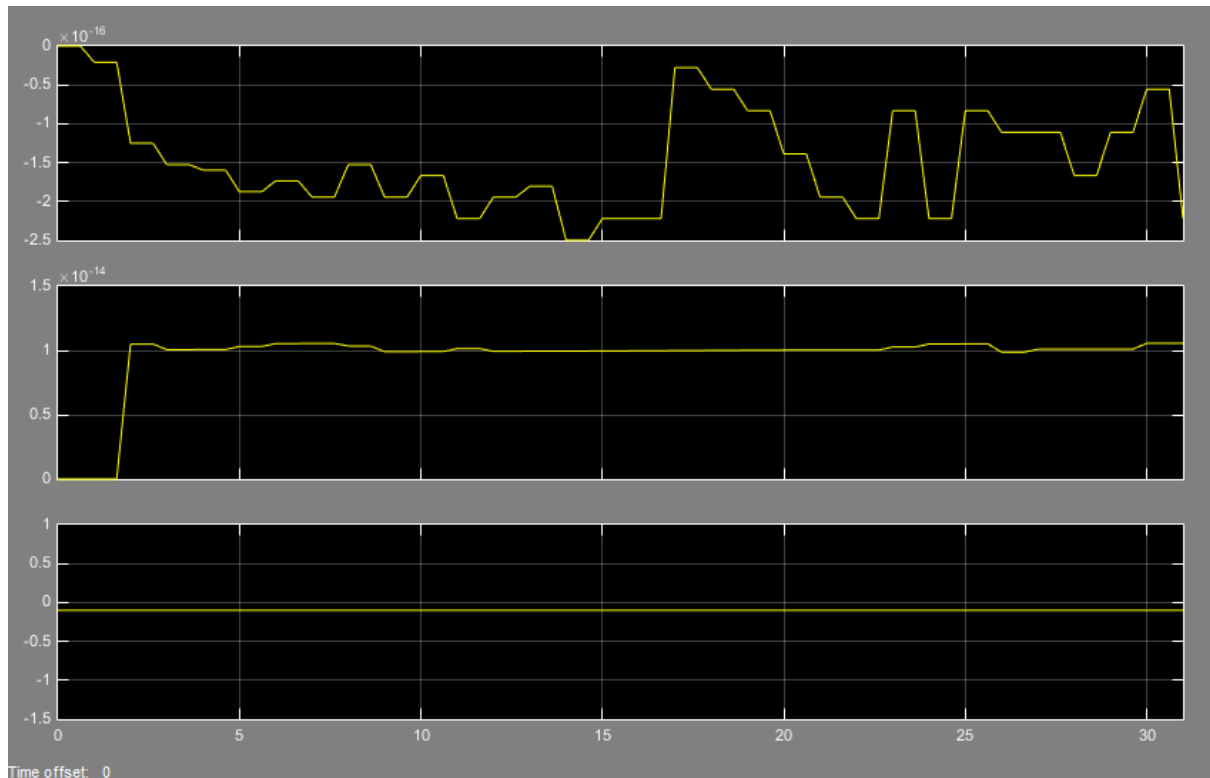


Figura 37 PI-A-Diferencia entre salidas

La trayectoria que describe el robot se puede ver dibujada en la siguiente imagen:

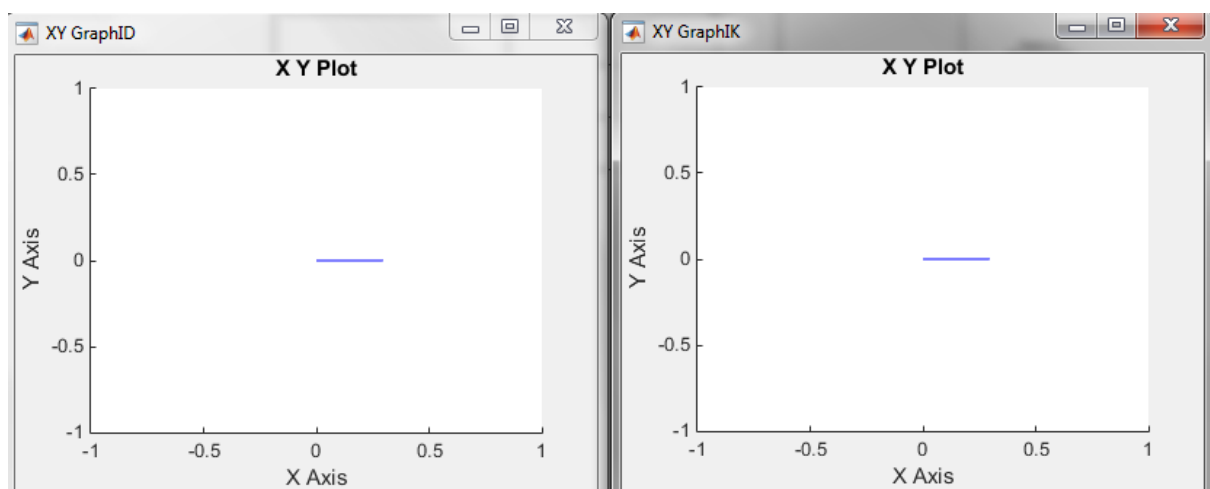


Figura 38 PI-A-Trayectoria dibujada IK vs.ID (m)

Salida del bloque cinemático inverso:



The figure consists of three vertically stacked plots, each showing the evolution of a specific coordinate over 30 iterations. The x-axis for all plots is labeled 'Iteraciones' and ranges from 0 to 30. The y-axis for the top two plots ranges from -1 to 1, and for the bottom plot, it ranges from 0 to 2.5.

- Top Plot (Eje X ID):** The y-axis is labeled 'Eje X ID'. The data is a constant horizontal line at y = 0.
- Middle Plot (Eje Y ID):** The y-axis is labeled 'Eje Y ID'. The data is a constant horizontal line at y = 0.
- Bottom Plot (Eje Z ID):** The y-axis is labeled 'Eje Z ID'. The data is a constant horizontal line at y = 1.1.

Figura 40 PI-B-Salida bloque ID (m/s)

Como se deduce de las gráficas anteriores no hay movimiento en ninguno de los ejes. Este resultado se debe al hecho de que la posición inicial del brazo es estirado en el eje Z por lo que al solicitar al robot moverse en dicho eje, no puede. La irregularidad de la gráfica IK para los ejes X e Y se debe al intento que hace el sistema por mover el brazo aunque como puede observarse en la gráfica de la figura 39, estas señales son tan pequeñas que no afectan al bloque dinámico inverso.

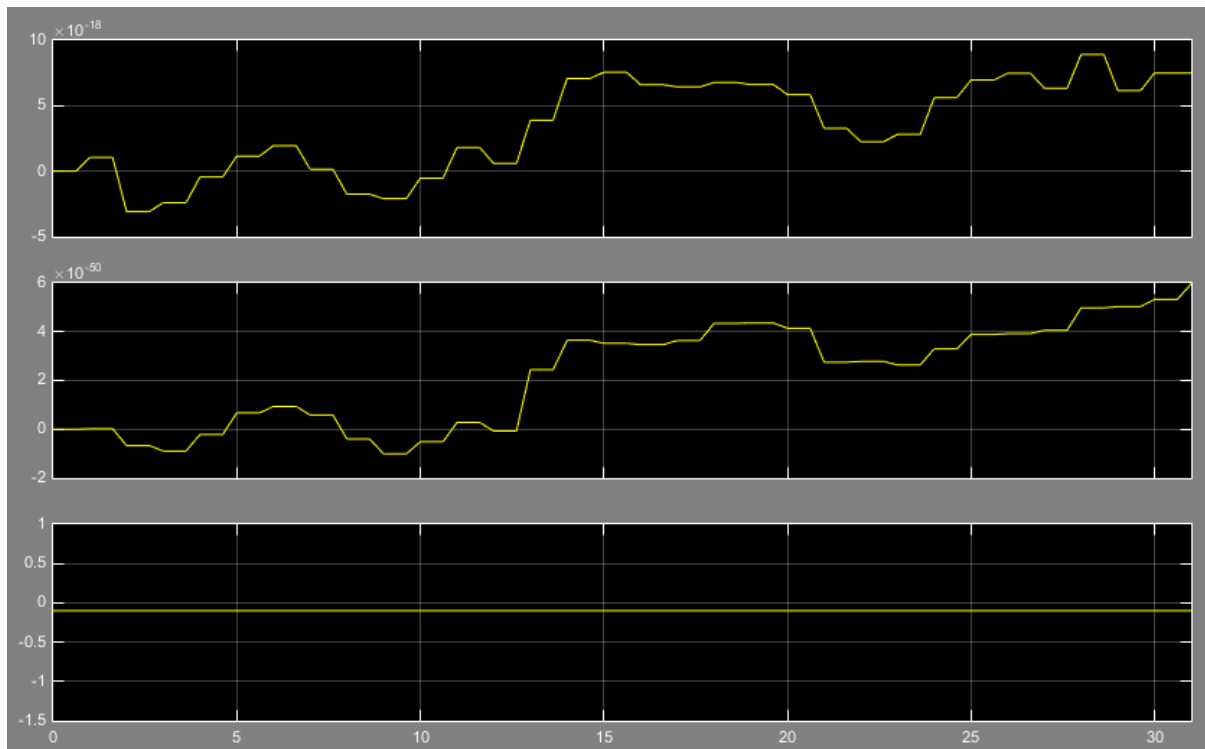


Figura 41 PI-B-Diferencia salidas (m/s)

Dado que no existe movimiento, no se dibuja ninguna trayectoria, como se muestra a continuación:

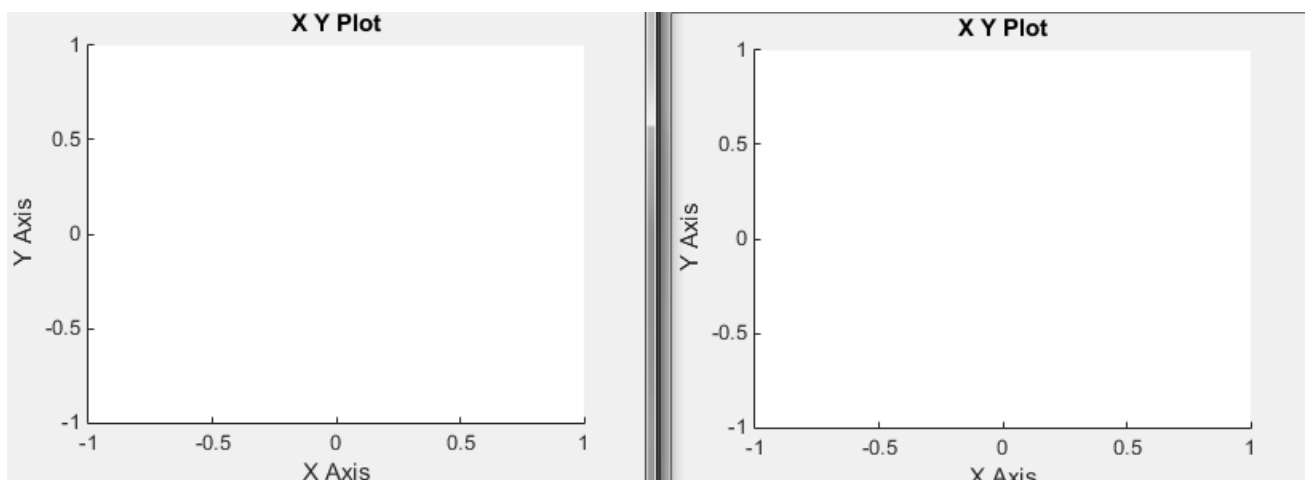


Figura 42 PI-B-Trayectoria dibujada IK vs ID(m)

4.2 PRUEBAS II: TRAYECTORIAS COMBINADAS

a- Movimiento continuo a velocidades $x=0.01\text{m/s}$, $y=0.01\text{m/s}$, $z=0\text{m/s}$

Salida bloque cinemático inverso:

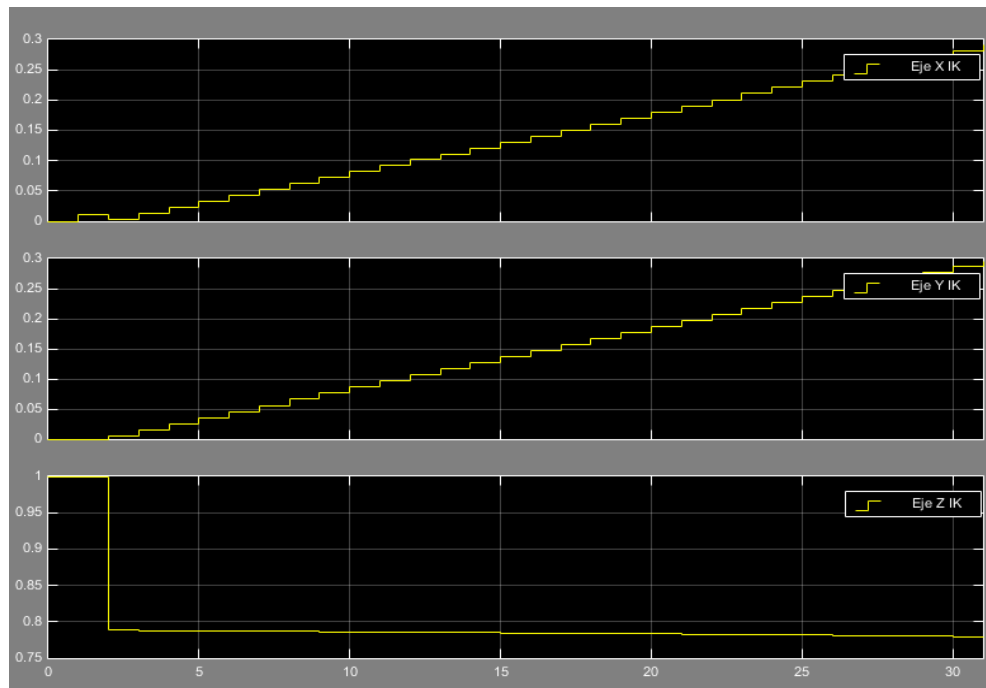


Figura 43 PII-A-Salida bloque IK (m/s)

Salida bloque dinámico inverso:

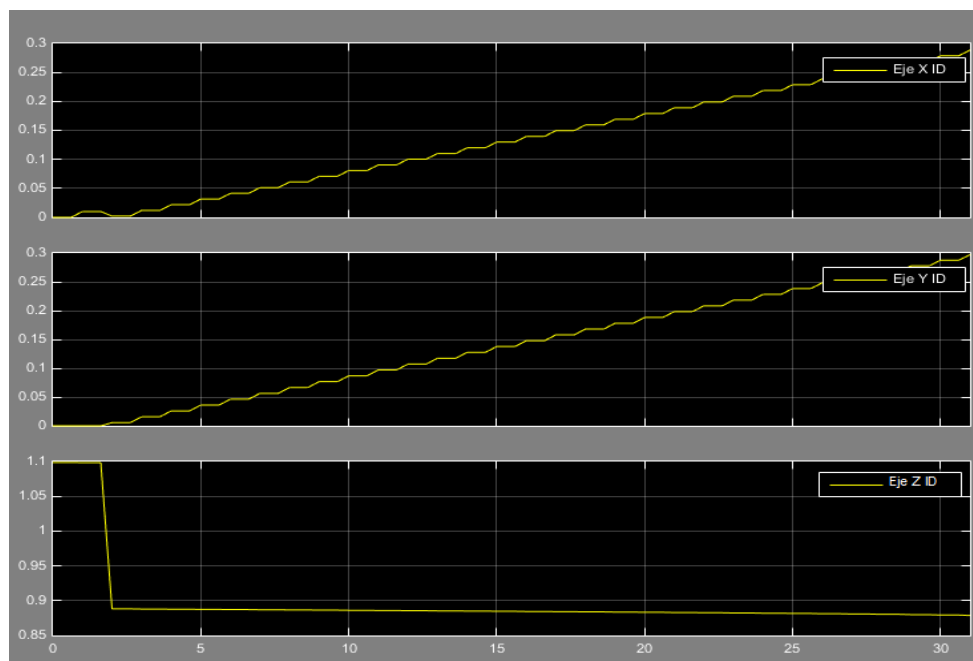


Figura 44 PII-A-Salida bloque ID (m/s)

Diferencia entre las salidas de ambos sistemas:

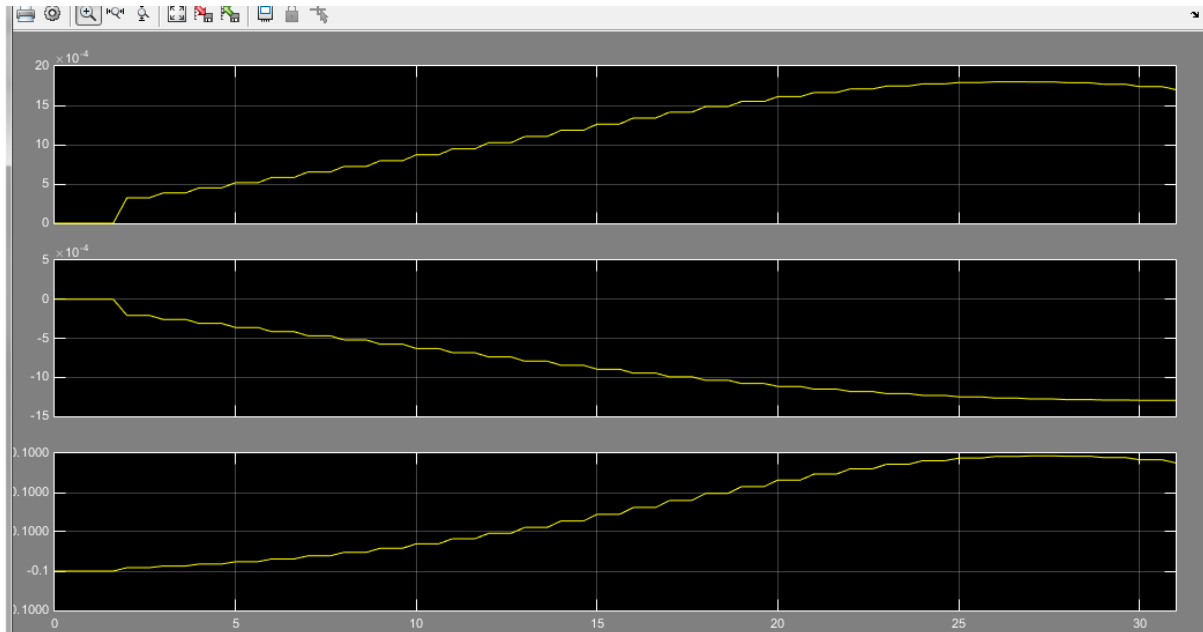


Figura 45 PII-A-Diferencia de salidas (m/s)

Trayectoria dibujada:

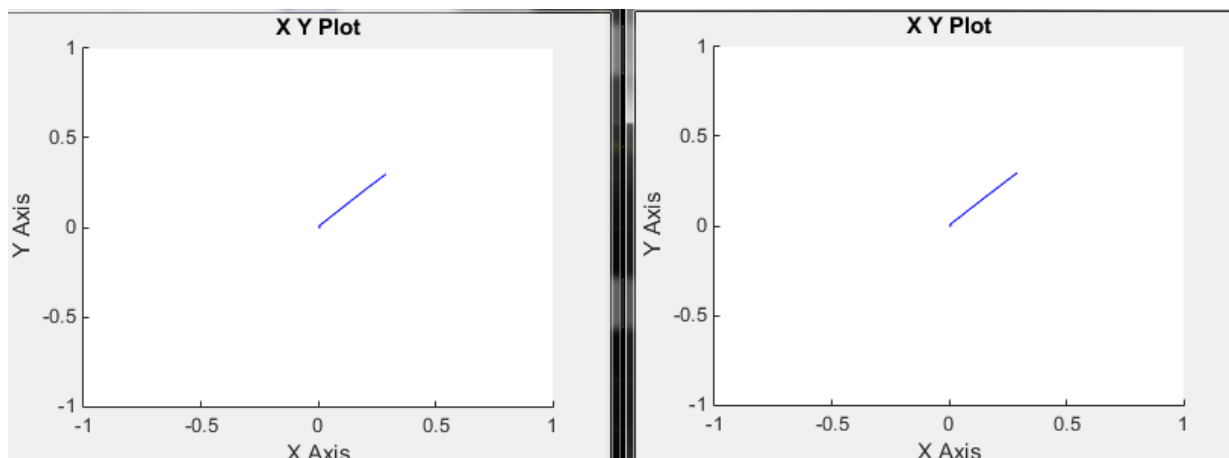


Figura 46 PII-A-Trayectoria dibujada IK vs. ID (m)

Como se puede observar la salida para ambos sistemas sigue siendo similar. Analizando la gráfica que resta ambos sistemas se podría pensar que las salidas no son iguales dadas las curvas existentes, pero hay que tener en cuenta que se tiene un error del orden de 1.10^{-4} por lo que puede considerarse nulo.

b- Movimiento continuo a velocidades $x=0.01\text{m/s}$, $y=0\text{ m/s}$, $z=0.01\text{m/s}$

Salida bloque cinemático inverso:

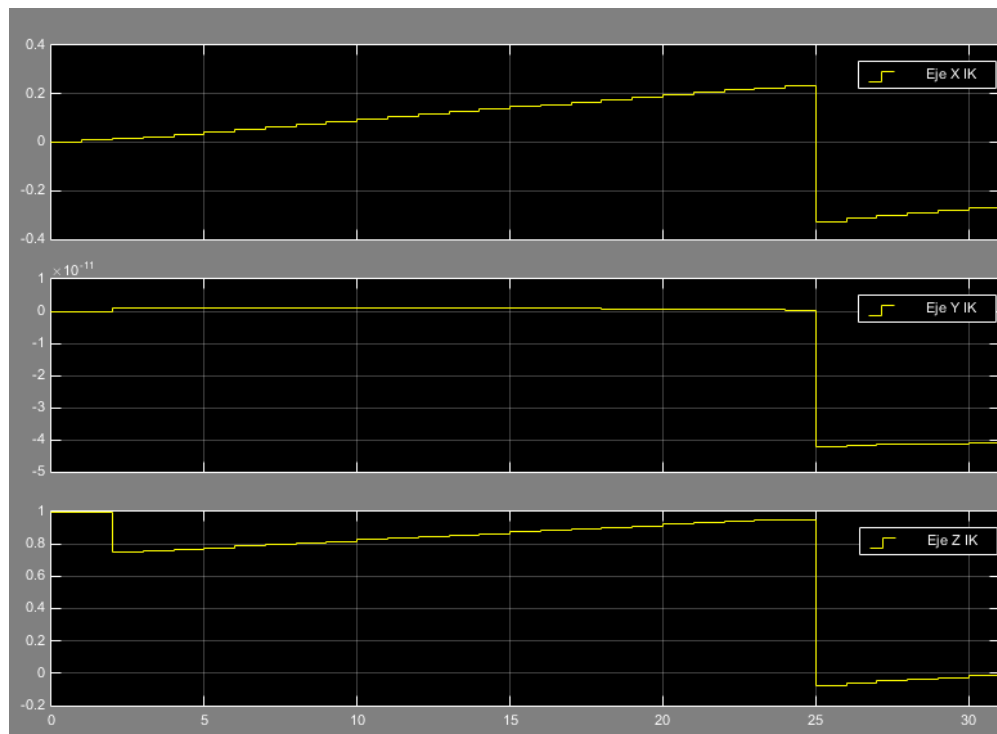


Figura 47 PII-B-Salida bloque IK (m/s)

Salida bloque dinámico inverso:

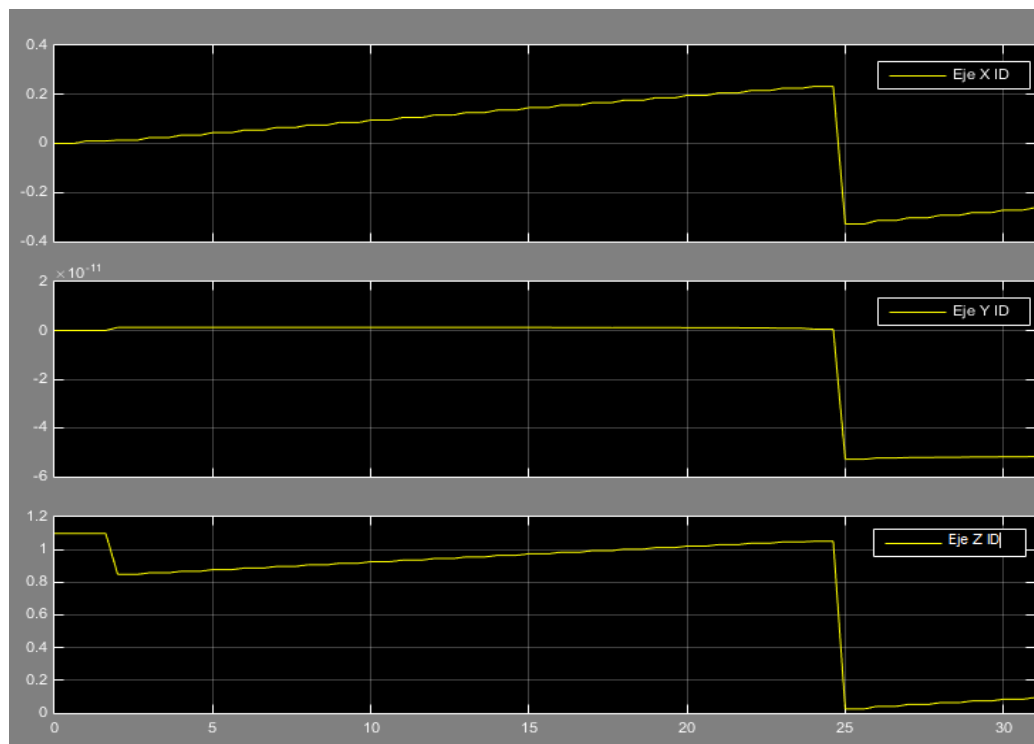


Figura 48 PII-B-Salida bloque ID (m/s)

Analizando ambas gráficas se observa un salto brusco cercano al segundo 25 de la simulación. Este salto se debe a que en esa franja de tiempo el brazo alcanza su longitud máxima por lo que para solventar este problema de su posición pero manteniendo el sentido de la trayectoria fijada.

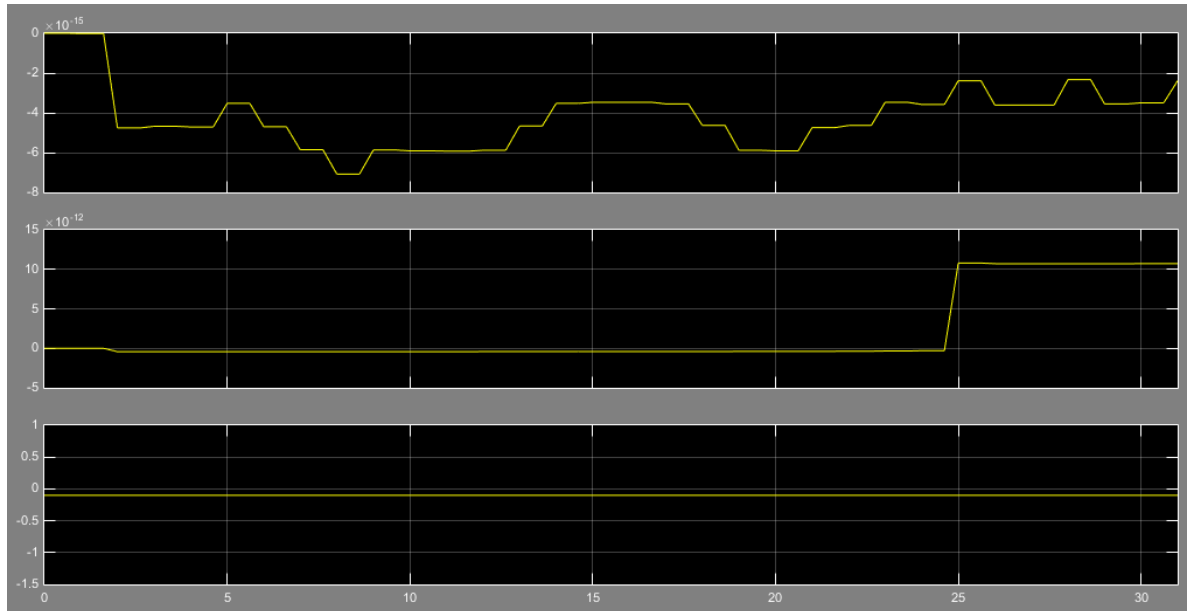


Figura 49 PII-B-Diferencia salidas (m/s)

Trayectoria dibujada por el brazo. Para facilitar la comprensión se ha decidido incluir dos gráficas correspondientes a los segundos 24 y 26 de la simulación para poder analizar el salto del brazo:

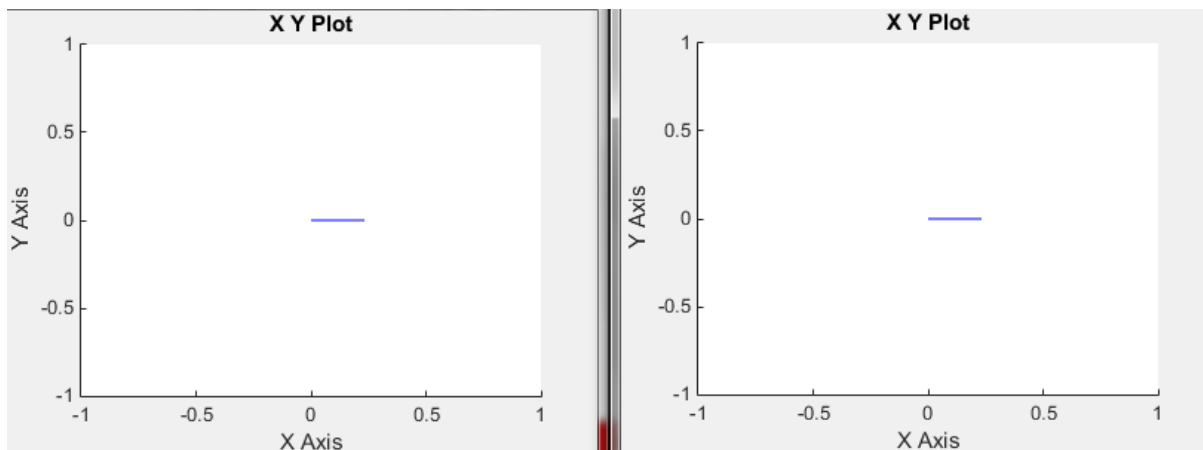


Figura 50 PII-B-Trayectoria dibujada hasta segundo 24 IK vs. ID (m)

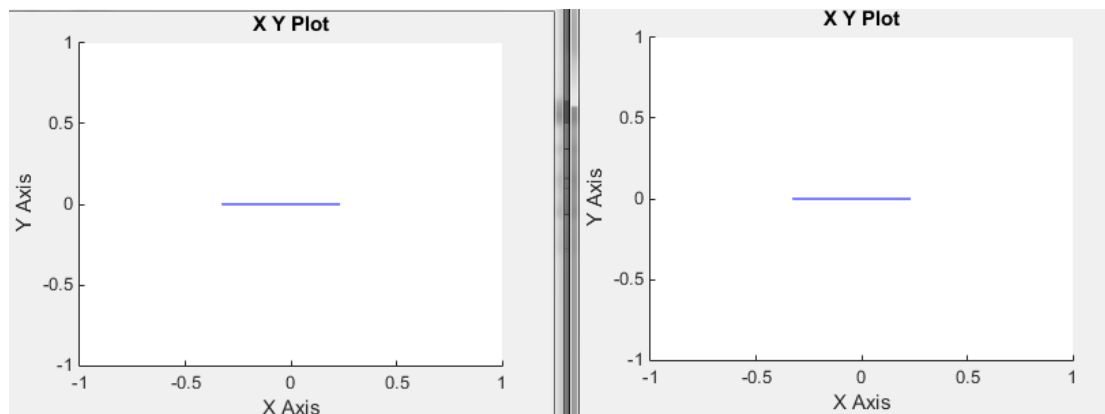


Figura 51 PII-B-Trayectoria dibujada tras segundo 24 IK vs ID(m)

Como se puede observar, el salto que hace el brazo para continuar con la trayectoria marcada hace que el robot retroceda y vuelva a iniciar su trayectoria. En la figura 50 la trayectoria comenzaba en el eje X en 0 mientras que en la figura 51 lo hace para $X=-0.3\text{m}$

c- Movimiento continuo a velocidades $x=0.01\text{m/s}$, $y=0.01\text{ m/s}$, $z=0.01\text{m/s}$

Salida bloque cinemático inverso:

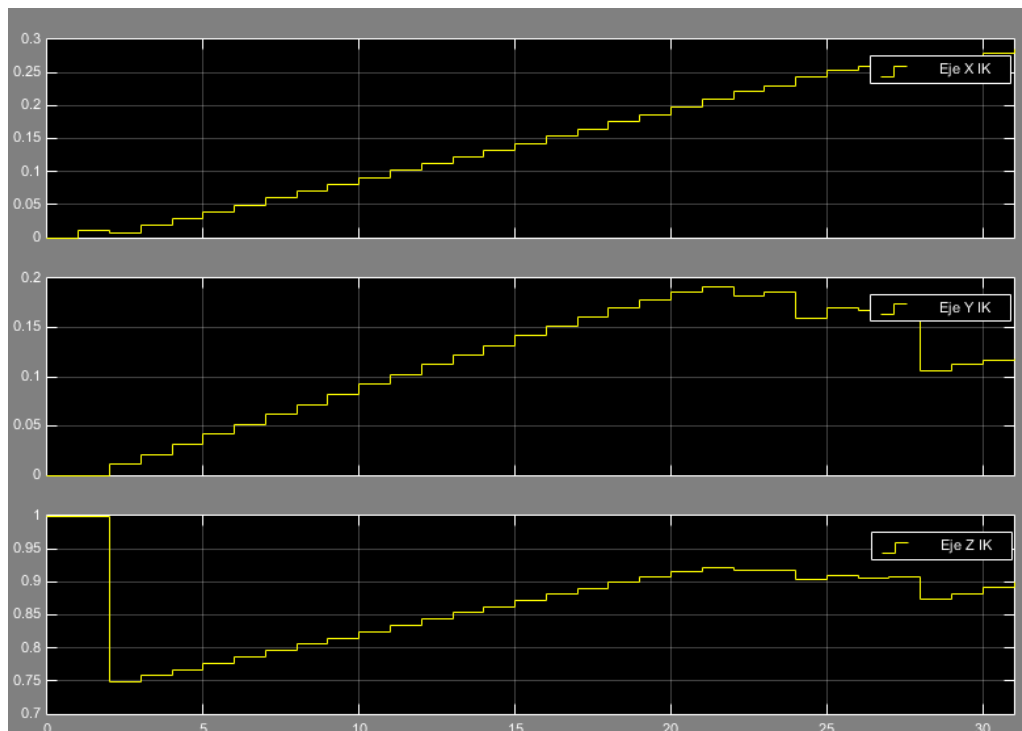


Figura 52 PII-C-Salida bloque IK (m/s)

Bloque dinámico inverso:

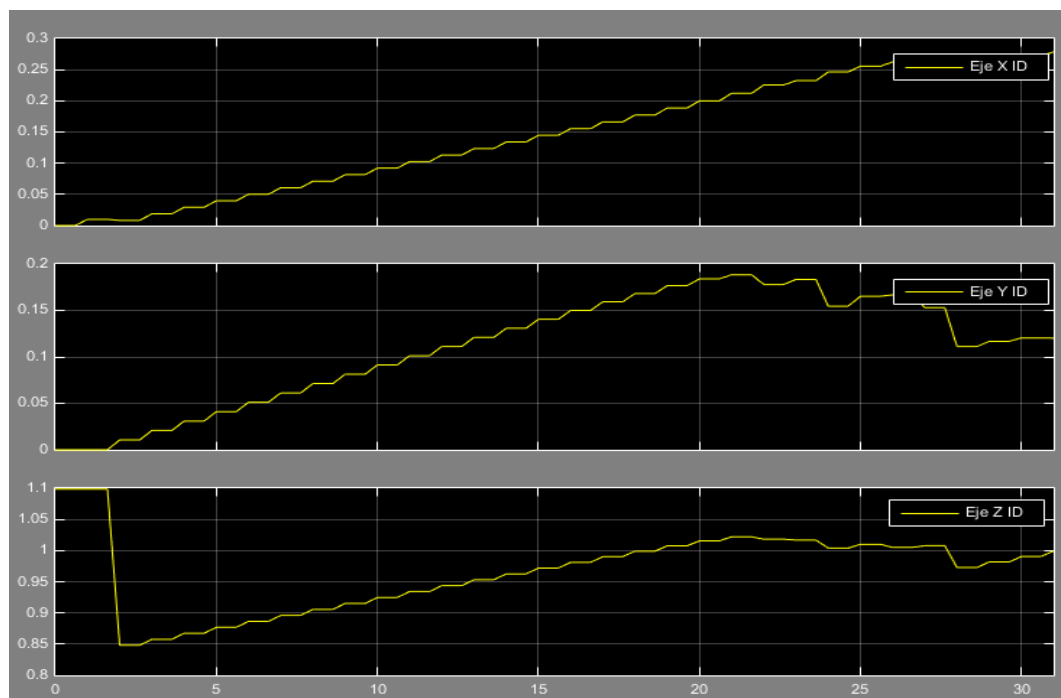


Figura 53 PII-C-Salida bloque ID (m/s)

Diferencia entre salidas (m/s):

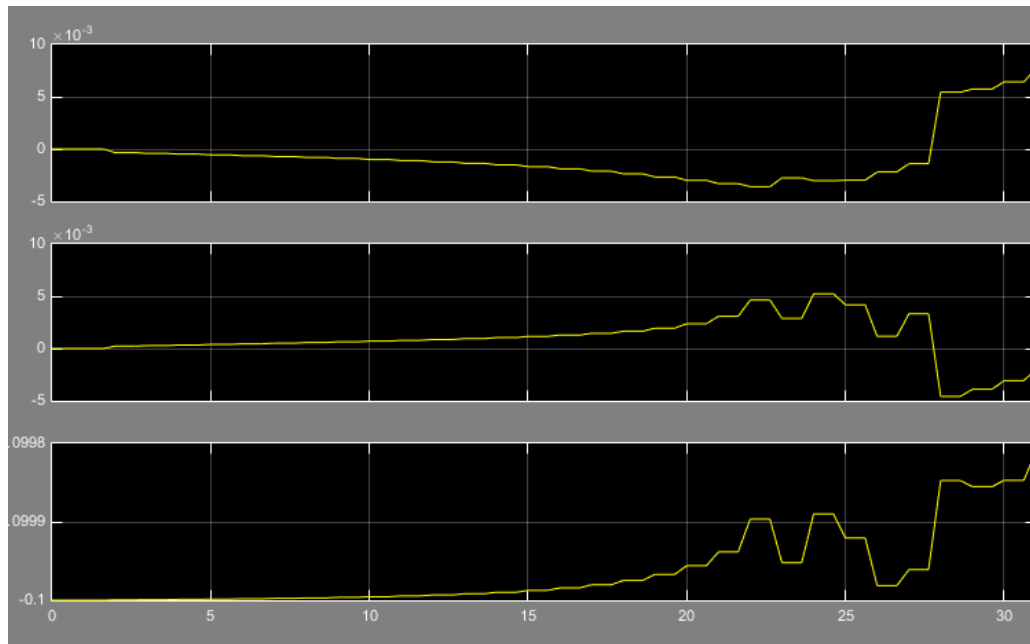


Figura 54 PII-C-Diferencia entre salidas (m/s)

Como se puede observar ambas salidas se mantienen similares hasta el segundo 20 de la simulación. En ese instante comienzan a producirse unas irregularidades que se hacen visibles en la trayectoria que dibuja el brazo robótico, como se observa en la figura siguiente:

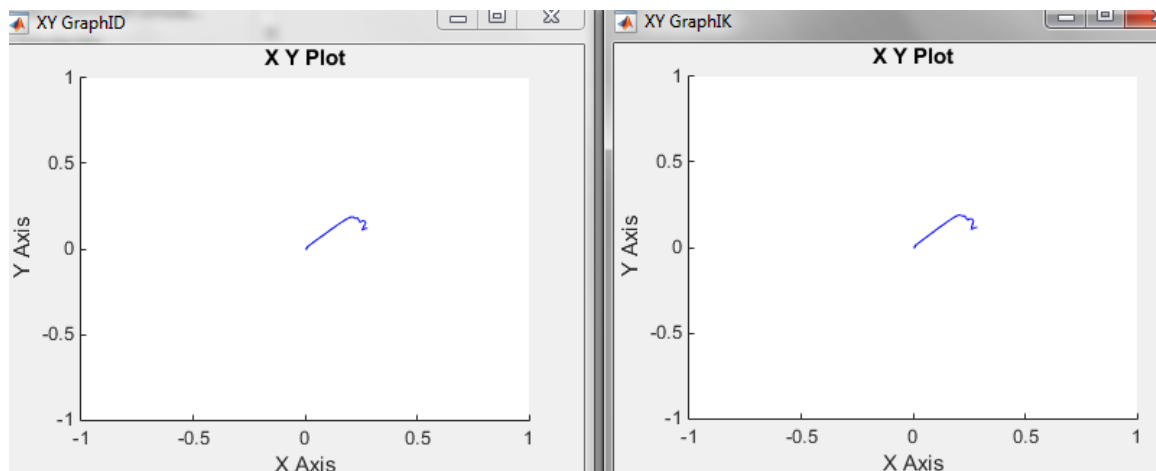


Figura 55 PII-C-Trayectoria dibujada IK vs.ID (m)

Igual que ocurría anteriormente, este error se debe a que en el segundo 20 el robot alcanza su posición de máximo estiramiento. En esta simulación, al movernos en los 3 ejes, el robot es incapaz de recuperar una posición desde la que seguir dibujando la trayectoria.

4.3 PRUEBAS III: TRAYECTORIA CIRCULAR

Para poder generar esta prueba ha sido necesario modificar el esquema de entrada al bloque cinemático inversor con el fin de definir el bloque de funciones que caracterizan un círculo. Este esquema puede encontrarse en ANEXO IV.

Para poder generar esta simulación es importante definir correctamente el punto inicial del brazo articulado, así como el centro del círculo a dibujar. Estos dos parámetros son muy importantes a la hora de determinar el movimiento del robot. Cualquier error en la determinación de estos valores puede dar lugar a la obtención de errores en la trayectoria y en los movimientos que el brazo puede o no realizar. Para definir la posición inicial del robot se ha utilizado el mismo proceso que quedó definido en el apartado 3.2.1.1 de este estudio. Para definir el centro del círculo se ha buscado un punto cercano a la posición inicial, con el fin de que el brazo no tenga que realizar ningún movimiento complejo evitando así los puntos singulares del brazo. Esta posición inicial se muestra en la figura siguiente:

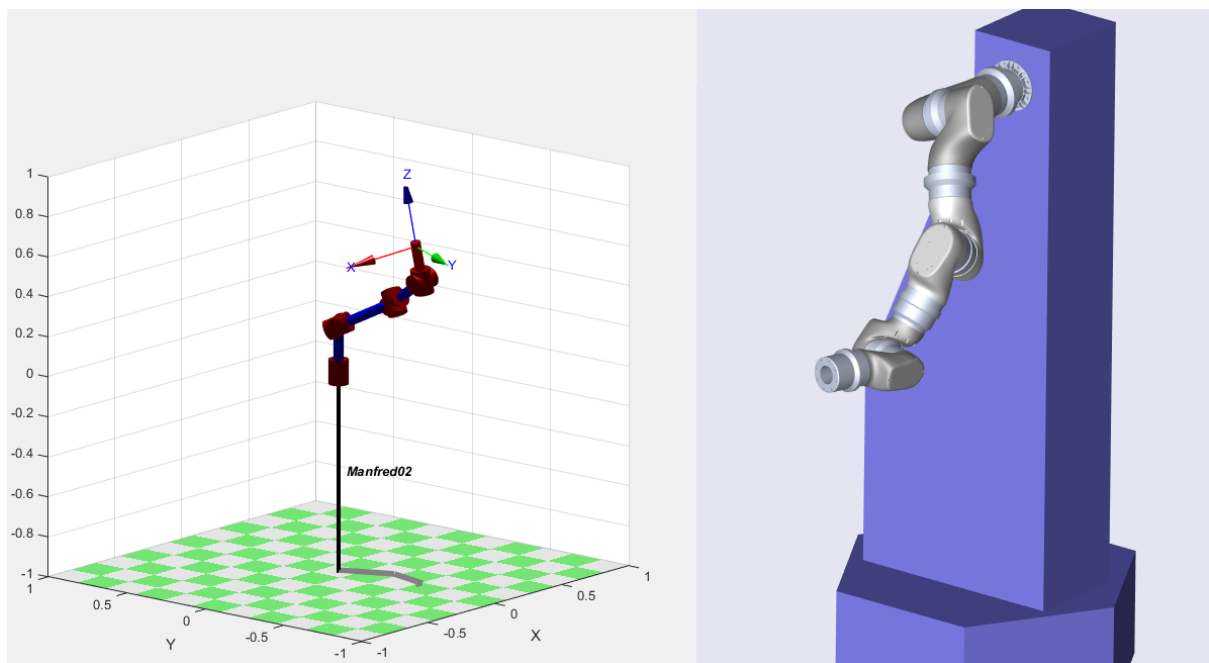


Figura 56 Posición inicial del robot previo dibujo de círculo

A continuación se muestran los resultados obtenidos de la simulación.

Bloque cinemático inverso (m/s):

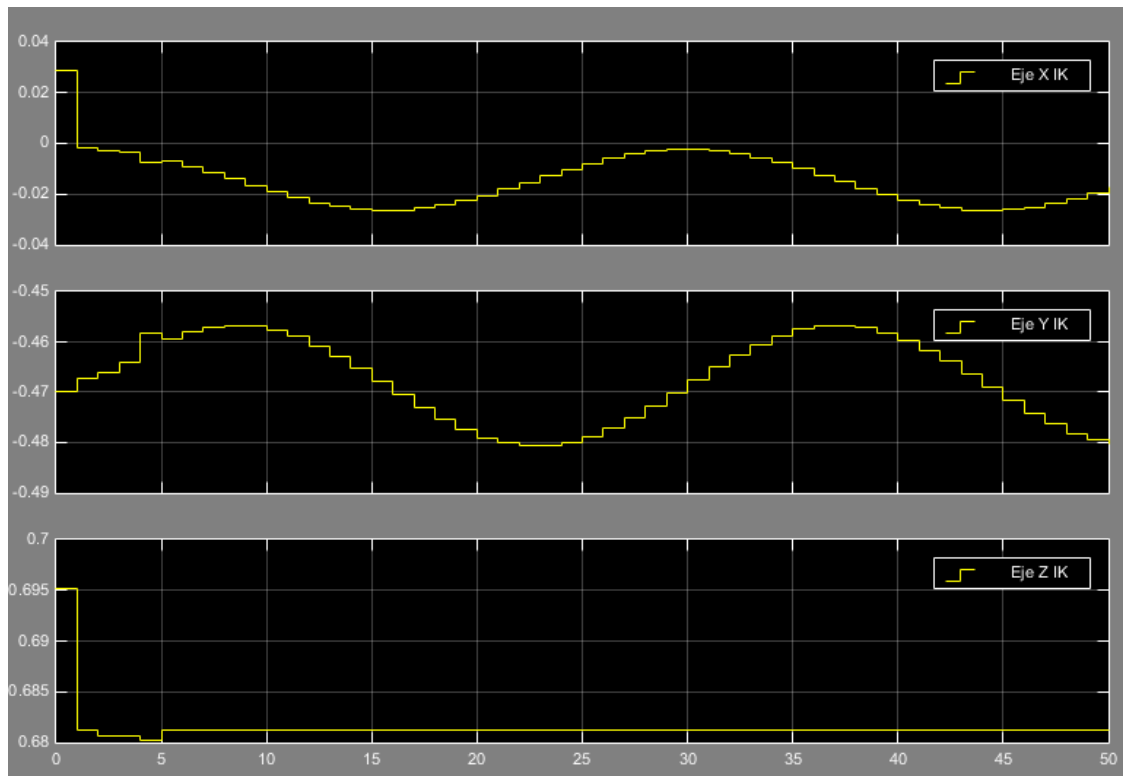


Figura 57 PIII- Bloque IK (m/s)

Bloque dinámico inverso (m/s):

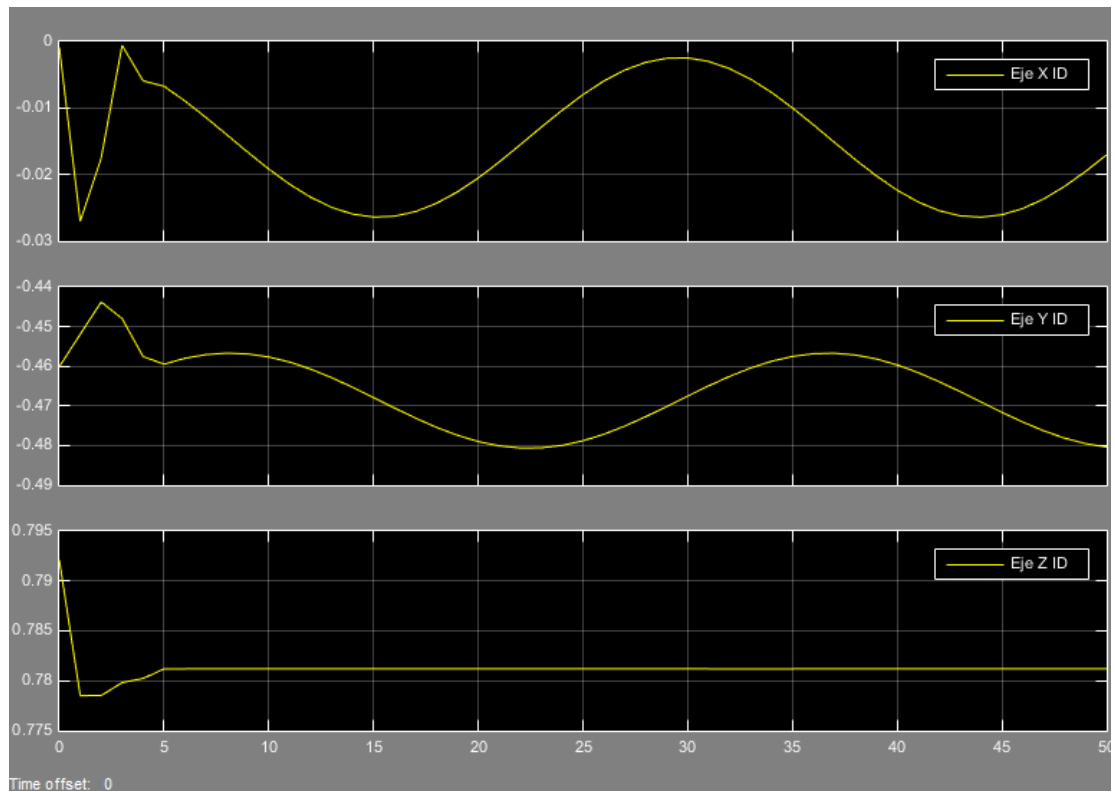


Figura 58 PIII-Bloque ID (m/s)

Diferencia entre salidas(m/s):

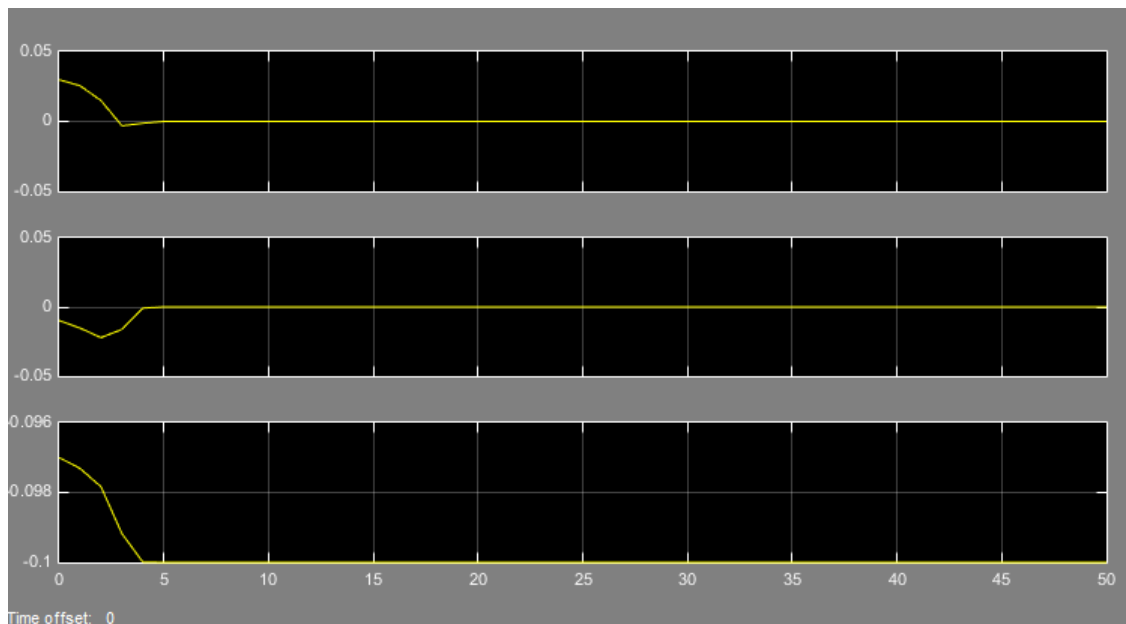


Figura 59 Diferencia salidas (m7s)

Dibujo de las trayectorias de salida IK vs.ID:

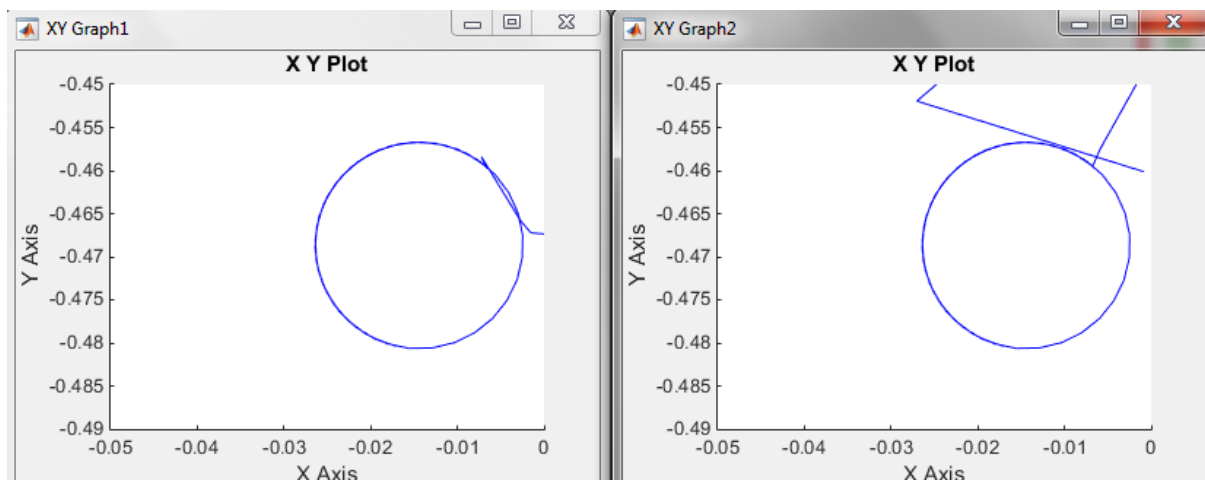


Figura 60 Trayectorias descritas por el brazo IK vs.ID

Como se puede observar hay una desviación entre ambos brazos durante los tres primeros segundos de la simulación, durante los cuales el brazo articulado se desplaza desde su posición inicial hasta la posición en la que comienza a dibujar el círculo. Además, durante la simulación de la trayectoria circular se ha modificado el bloque dinámico inverso, incluyendo un sensor a cada motor, o bloque *revolute* acorde al diagrama. Dado que la dinámica inversa tiene en cuenta el peso y momentos de inercia del brazo articulado, mediante este sensor se puede obtener el par asociado a cada motor. Los resultados obtenidos han sido coherentes, entrando dentro del rango de valores que los motores pueden alcanzar. Estos datos se muestran en la siguiente gráfica:

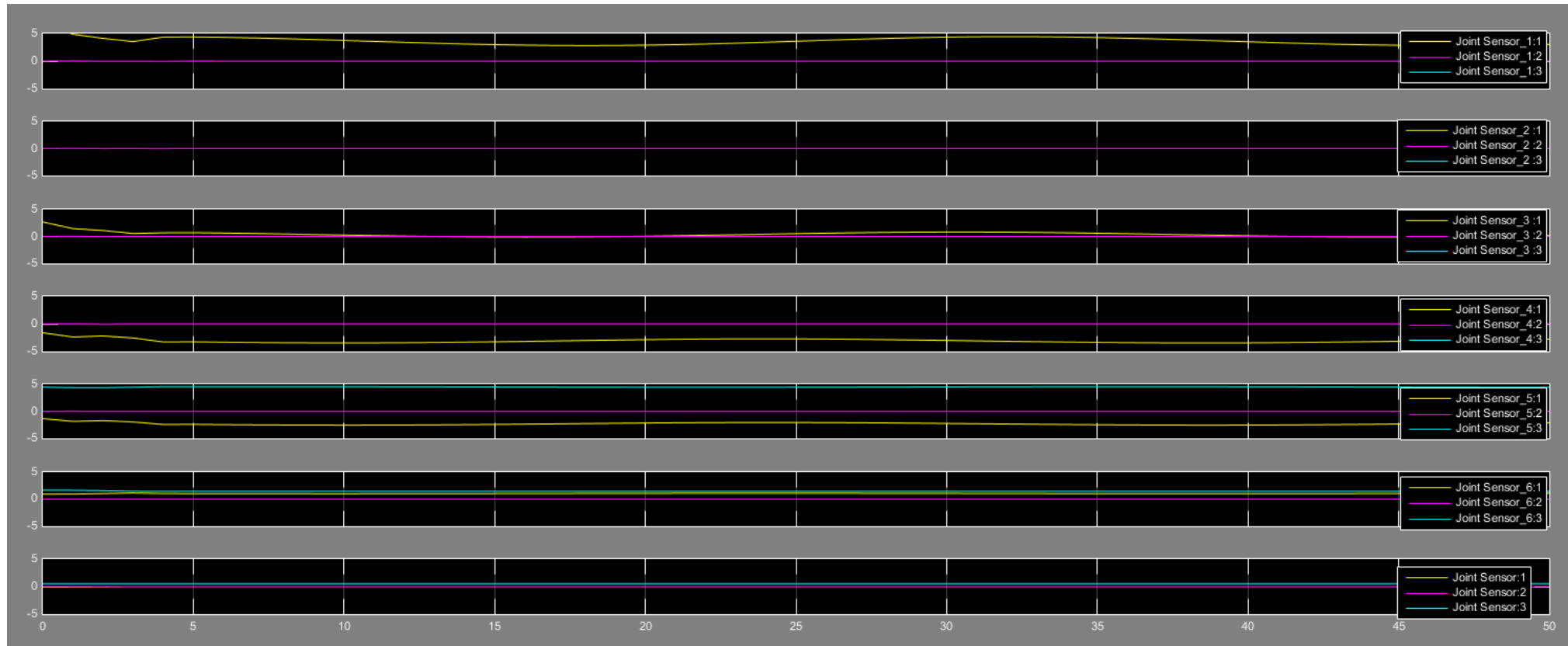


Figura 61 Par motor obtenido



5 CONCLUSIONES

En base a las pruebas realizadas se puede concluir que se ha conseguido generar un primer entorno de simulación virtual estable para el MANFRED 3, sobre la que realizar pruebas de trayectorias y análisis del par de los motores y que puede ser utilizado como base en futuros estudios. Se abren así numerosas posibilidades para continuar con el desarrollo de este trabajo. Quizá el objetivo más fácil de alcanzar a corto plazo es obtener una simulación de la dinámica directa del manipulador, para así poder compararla con los resultados obtenidos del bloque dinámico inverso, en cuanto al análisis del par motor se refiere. También sería interesante realizar pruebas sobre movimientos con velocidad rotacional ya que las aquí realizadas no han tenido en cuenta este factor.

Del mismo modo existen varios puntos de mejora en este proyecto como son:

- Modificación de la función Jacobiana: Se podría incluir en la función un segundo bloque de ajuste identificado en el manual de robótica, [7] con el nombre de función NULL, o control del espacio nulo. Mediante esta función evitamos que una articulación esté cerca de su límite mecánico. De este modo podemos solicitar al brazo que realice una trayectoria determinada mientras que a una de las articulaciones se le solicita alejarse de dicho límite
- Ángulos límite en modelado SimMechanics.- Podrían incluirse limitaciones de movimiento que mejorarían el comportamiento del brazo articulado, impidiendo posibles colisiones con la plataforma del brazo y posibles errores en la simulación. Una solución aproximada se consigue mediante la inclusión de muelles amortiguadores. De cualquier modo este trabajo no es del todo sencillo y realizando la simulación dinámica mediante la inclusión de los datos del torque de los motores quedaría resuelto.

Tanto el ajuste del entorno de simulación como las pruebas realizadas requieren de una buena comprensión del trabajo que se está realizando, además de probar en repetidas ocasiones las simulaciones, aplicando ligeros cambios entre ellas, con el fin de conseguir entornos favorables en los que poder realizar las pruebas. Este hecho ha dado lugar a que sea de vital importancia el soporte técnico facilitado por los tutores de este estudio en este punto del proyecto, lo cual no quería finalizar sin mencionarlo.

BIBLIOGRAFÍA

- [1] Mena, R; Flores. A; Copaci. D; Álvarez. D. "Design and development of a robotic arm for MANFRED 3 using low weight joints"; Descripción de diseño para el Proyecto de construcción del brazo para MANFRED 3. Documento técnico interno RoboticLab, 2014.
- [2] Proyecto MANFRED: <http://roboticslab.uc3m.es/roboticslab/robot/manfred-2>; Última visita 10 de Septiembre de 2015
- [3] SolidWorks Education. "Guía del estudiante para el aprendizaje del software SolidWorks". PMS0118-ESP
- [4] Simulink User's Guide. Revised for Simulink 8.6, September 2015 (Online only)
- [5] SimMechanics™ User's Guide, Revised for Version 2.7, September 2007 (Online only)
- [6] SW Peter Corke Robotics Toolbox: http://www.petercorke.com/Robotics_Toolbox.html; Última visita, 19 de Septiembre de 2015
- [7] Corke, P. "Robotics, Vision and Control: Fundamental Algorithms in MATLAB (Springer Tracts in Advance Robotics)". Ed. Springer, 2011. ISBN:978-3-642-20143-1
- [8] Corke, P. "Robotics Toolbox for MATLAB. Release 9". Release 9.10, Febrero 2015
- [9] Robotics Lab: <http://roboticslab.uc3m.es/roboticslab/>; última visita Martes 1 de Septiembre de 2015
- [10] Blanco D., Al Ansari S., Castejón C., López Boada B., Moreno L.E. "MANFRED: Robot antropomórfico de servicio fiable y seguro para operar en entornos humanos" en Revista Iberoamericana de Ingeniería Mecánica. Vol. 9, Nº 3, pp. 33-48, 2005
- [11] Rodríguez Pérez C.A. "Diseño de un robot antropomórfico de propósito general", Universidad Tecnológica de Pereira. Facultad de Tecnologías. Ingeniería Mecatrónica por Ciclos. 2013



ANEXOS

ANEXO I: PRESUPUESTO

Con objeto de realizar una estimación lo más verosímil posible de los costes económicos que suponen este proyecto se dividirá el capítulo en 3 apartados:

- Desglose de horas
- Coste de materiales
- Costes de personal cualificado
- Coste total

DESGLOSE DE HORAS

Con el fin de obtener una estimación del tiempo de realización del presente trabajo se hará un desglose de las horas invertidas por el autor del estudio según las fases que comprenden este proyecto.

1ª FASE: Familiarización:

- Conocimiento del brazo antropomórfico MANFRED: 5 horas.

2ª FASE: Herramientas SW:

- Instalación de los entornos de diseño y simulación: 2 horas.
- Familiarización y tutorial de SolidWorks: 7 horas.
- Familiarización entorno y tutorial de Matlab-SimMechanics: 10 horas.
- Familiarización entorno y estudio manuales de la *toolbox* Peter Corke: 25 horas.

3ª FASE: Desarrollo del estudio:

- Diseño y ensamblaje del brazo robótico en SolidWorks: 10 horas.
- Diseño del brazo con SimMechanics: 5 horas.
- Diseño del brazo con la *toolbox* Matlab-Peter Corke: 4 horas.
- Diseño de la simulación (Peter Corke + SimMechanics): 120 horas.
- Pruebas de simulación tras generar el entorno adecuado: 5 horas
- Redacción de la memoria: 85 horas

TOTAL HORAS INVERTIDAS: 278 h

COSTES DE MATERIAL

Para la consecución de este proyecto ha sido necesario la adquisición de equipos y licencias SW que se enumeran en la siguiente tabla.

Tabla 3 Costes de material

Código	Unidad	Descripción	Cantidad	Coste unidad (IVA incluido)	Coste Total
010	PC	Ordenador de trabajo	1	500,00€	500,00€
020	Herramienta SW	Licencia Standard de SW SolidWorks 2014	1	6000,00€	6000,00€
030	Herramienta SW	Licencia de SW Matlab	1	2000,00€	9000,00€
		-Complementos:		-Complementos:	
		○ Simulink	1	3000,00€	
		○ Simscape	1	2000,00€	
		○ SimMechanics	1	2000,00€	
040	Herramienta SW	Licencia de SW Peter Corke Toolbox(Descarga libre)	1	0,00€	0,00€
TOTAL COSTES MATERIALES: 15500,00 €					

COSTES DE PERSONAL

En la siguiente tabla se muestra el personal necesario para la realización del proyecto:

Tabla 4 Costes de personal

Código	Unidad	Descripción	Cantidad	Horas invertidas en el estudio	Coste unidad/hora (IVA incluido)	Coste Total
110	Ingeniero Electrónico	Ingeniero en Electrónica y automática encargado del desarrollo del estudio: SIMULACIÓN DINÁMICA DE UN BRAZO ROBÓTICO EN SIMMECHANICS	1	278	15€	4170€
120	Ingeniero	Asesor Técnico	2	90 h/asesor	12,5	2250€

TOTAL COSTE PERSONAL: 6420 €

COSTE TOTAL

En base al desglose descrito en los apartados previos del presente capítulo se puede generar una tabla que refleje los costes totales necesarios para el desarrollo del estudio objeto de esta memoria.

Tabla 5 Coste total

CONCEPTO	PRECIO (€)
Costes de material	15500,00€
Costes de personal	6420,00€
TOTAL	21920,0 €

A partir de la tabla de costes totales, se concluye que el coste del proyecto ha sido de:

Veintiún mil novecientos veinte euros.

ANEXO II: PROGRAMACIÓN DEL BRAZO ANTROPOMÓRFICO

CÓDIGO DE DEFINICIÓN DEL BRAZO

La siguiente línea de comandos define el brazo articulado:

```
>>deg=pi/180;  
>>L (1) = Revolute ('d', 0.228, 'a', 0, 'alpha', pi/2,'qlim', [-100 100]*deg);  
>> L (2) = Revolute ('d', 0, 'a', 0, 'alpha', pi/2,'qlim', [-100 100]*deg);  
>>L (3) = Revolute ('d', 0.3125, 'a', 0, 'alpha', -pi/2,'qlim', [-170 170]*deg);  
>>L (4) = Revolute ('d', 0, 'a', 0, 'alpha', pi/2,'qlim', [-100 100]*deg);  
>>L (5) = Revolute ('d', 0.278, 'a', 0, 'alpha', -pi/2,'qlim', [-170 170]*deg);  
>>L (6) = Revolute ('d', 0, 'a', 0, 'alpha', pi/2,'qlim', [-100 100]*deg);  
>>L (7) = Revolute ('d', 0.18, 'a', 0, 'alpha', 0,'qlim', [-170 170]*deg);  
>>L  
>>Manfred02 = SerialLink (L, 'name', 'Manf02');
```

OTROS COMANDOS ÚTILES

```
>>Manfred02. plot() : //Genera es esquema gráfico del brazo  
>>Manfred02. teach() : // Permite modificar desde el esquema gráfico la posición del brazo  
>>Manfred02. getpos() : // Muestra la posición del brazo  
>>cg = CodeGenerator (Manfred02) : //Genera el código asociado al brazo  
>> cg. geneverything() : //Genera todos los bloques Simulink asociados al brazo
```



ANEXO III: PIEZAS SOLIDWORKS

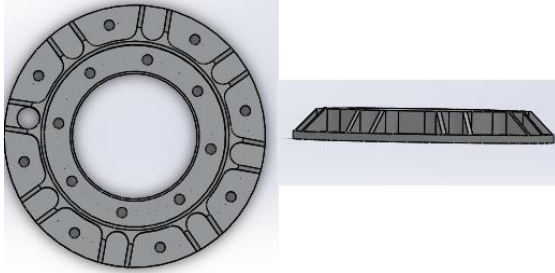


Figura 62 12012_ARQ_M3_A003_P_19_00

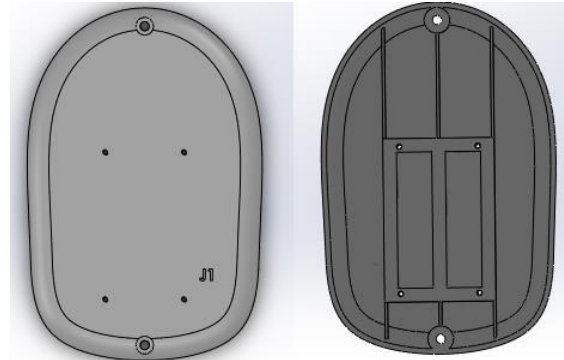


Figura 66 12102_ARQ_M3_A003_P_00J1

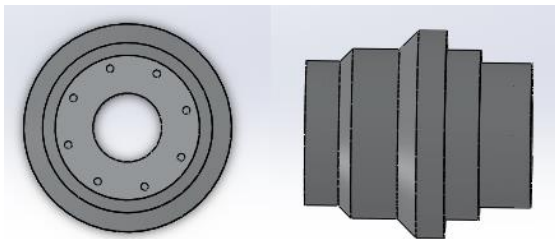


Figura 63 Dummy_J1F-12

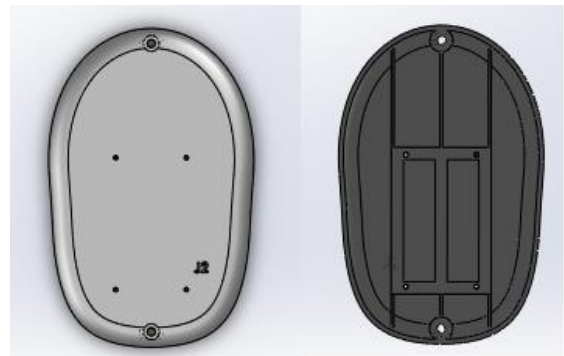


Figura 67 12102_ARQ_M3_A003_P_00J2

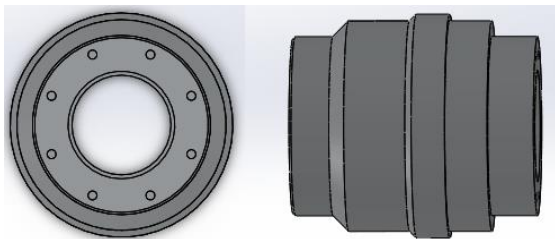


Figura 64 Dummy_J2F_12

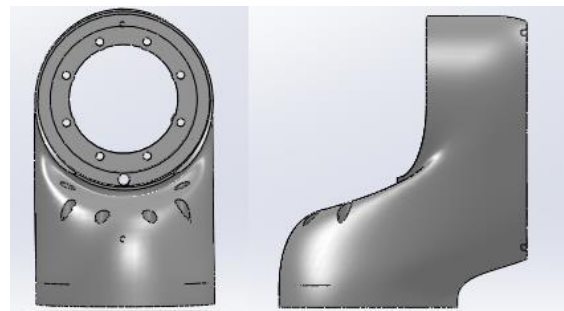


Figura 68 12102_ARQ_M3_A003_P_00D2

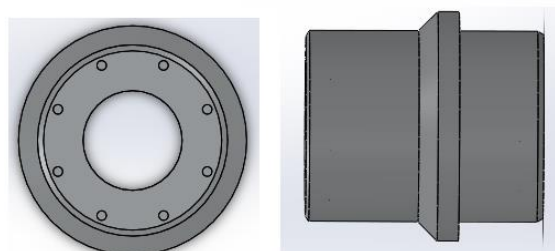


Figura 65 Dummy_J3F_12

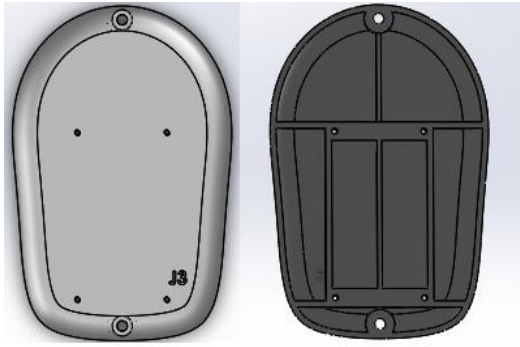


Figura 69 12102_ARQ_M3_A003_P_00J3

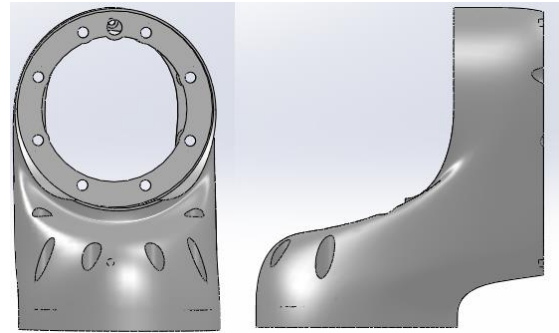


Figura 71 12102_ARQ_M3_A003_P_00D3

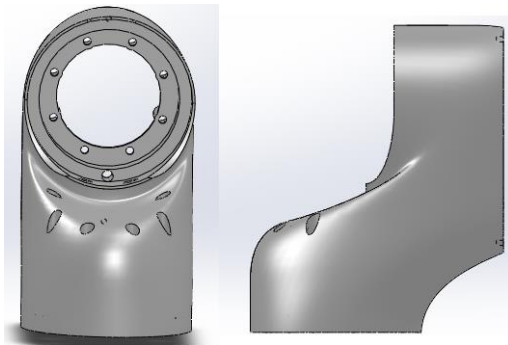


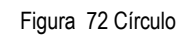
Figura 70 12102_ARQ_M3_A003_P_00D1



ANEXO IV: ESQUEMAS DE ENSAMBLAJES

En el siguiente anexo se mostrarán algunos de los ensamblajes generados a lo largo de la simulación. Entre otros se podrán encontrar:

- Bloque Círculo
- Esquema completo de la simulación



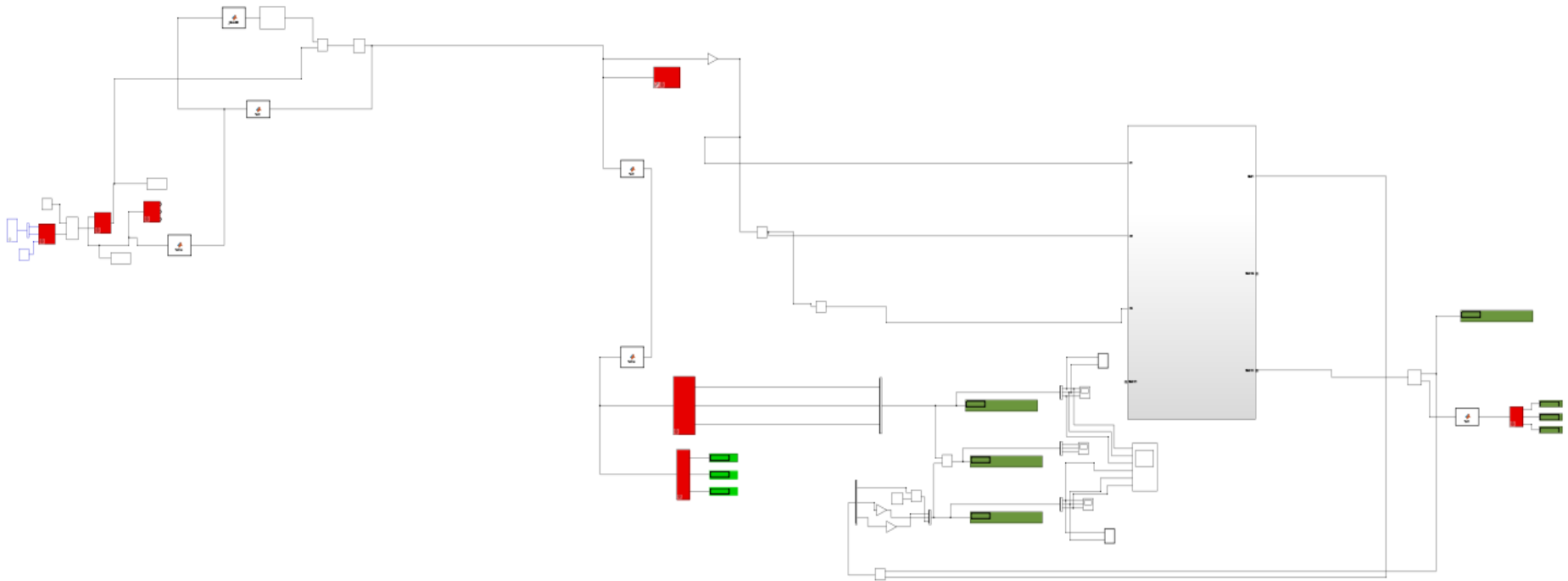


Figura 73 Esquema Final del entorno de simulación

